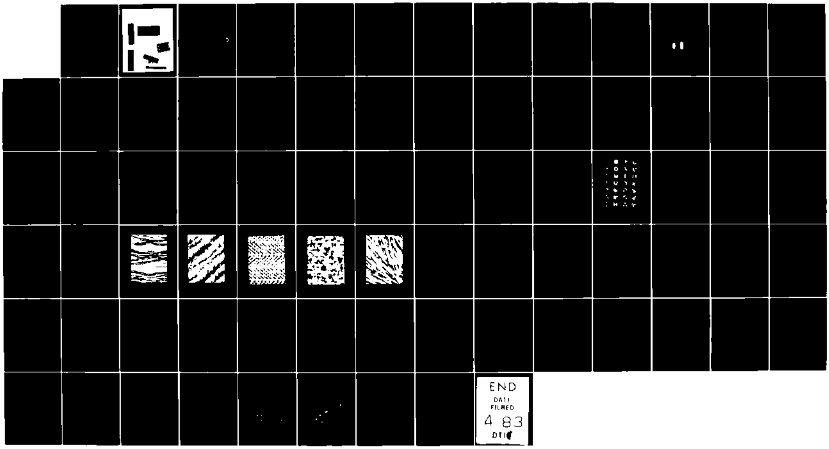
AD-A125 710 AUTOMATED ANALYSIS OF IMAGERY(U) INTERACTIVE SYSTEMS CORP LITTLETON CO D LUCAS ET AL. NOV 82 AFOSR-TR-83-0055 F49620-82-C-0070 1/1

UNCLASSIFIED F/G 20/6 NL

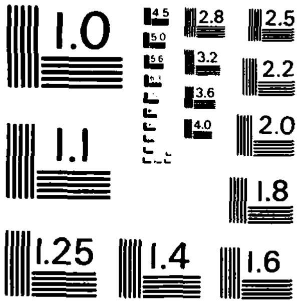


END

DATE FILMED

4 83

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFOSR-TR- 83-0055 (12)

Automated Analysis of Imagery

AFOSR-TR- 83-0055 12

Automated Analysis of Imagery

Dean Lucas
Laurie Gibson

Interactive Systems Corporation

November 1982

Final Report for Period 1 April 1982 - 1 October 1982

Prepared for:

THE AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Building 410
Bolling Air Force Base, D.C. 20332

Approved for public release;
distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83-0055	2. GOVT ACCESSION NO. A125 710	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Automated Analysis of Imagery		5. TYPE OF REPORT & PERIOD COVERED Final - 1 Apr 82 to 1 Oct 82
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dean Lucas Laurie Gibson		8. CONTRACT OR GRANT NUMBER(s) F49620-82-C-0070
9. PERFORMING ORGANIZATION NAME AND ADDRESS Interactive Systems Corporation 5500 South Sycamore Street Littleton, Colorado 80120		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2305/K1
11. CONTROLLING OFFICE NAME AND ADDRESS USAF,AFSC Air Force Office of Scientific Research/NE Building 410, Bolling AFB, D.C. 20332 Code: FQ8671		12. REPORT DATE November 1982
		13. NUMBER OF PAGES 75
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release;
distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image analysis; texture; shape descriptors; Generalized Balanced Ternary

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objective of this research was to study possible techniques for extracting features and textural regions from imagery using Generalized Balanced Ternary. In addition, various aggregate measures were defined and their properties investigated as to their applicability in representing features. A mathematical definition of texture was derived based on the variance of the picture function over areas. This definition was tested on sample textures. Among

(OVER)

20. the aggregate measures that were derived and studied was a function, f^2, which distinguishes shapes. f^2 is invariant under size, rotation, translation, and scaling in any axis. This measure and others were used in software which converted images in pixel form to higher level representations.

TABLE OF CONTENTS

Section Number — Description — Page

1.0 INTRODUCTION 1

2.0 STATEMENT OF THE PROBLEM 2

3.0 SCOPE 3

4.0 DESCRIPTION OF THE TECHNOLOGY 4
4.1 Picture Functions 4
4.2 Classical Techniques 5
4.3 Examples 6
4.4 f and f^2 11
4.5 Properties of f and f^2 17
4.6 Generalized Balanced Ternary (GBT) 22
4.7 G and H Functions 25
4.8 Agglomeration 27
4.9 More Examples 28
4.10 Texture 32

5.0 IMPLEMENTATION 35
5.1 Texture 35
5.2 Shape Descriptors 42

6.0 FUTURE RESEARCH DIRECTIONS 50
6.1 Texture 50
6.2 Shape 51

7.0 BIBLIOGRAPHY 53

8.0 APPENDIX. GENERALIZED BALANCED TERNARY 54

LIST OF FIGURES

1 An Illustration of a Binary Picture Function 4
2 A Binary Picture Function of a Rectangle centered at the Origin together with significant Statistics 6
3 The Comparable Picture Function and Statistics for an Ellipse centered at the Origin with Semiaxes a and b 7
4 The Comparable Picture Function and Statistics for a Rhombus centered at the Origin with Semiaxes a and b 7
5 The Basic Triangle, T, from which an n-gon is constructed 7
6 N identical rectangles aligned and equally spaced. Rectangles are of height 2h and width w $(0 \leq w \leq 1)$ 10

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

Number	Description	Page
7	How Translation, Rotation, and Scaling are accomplished by Addition and Multiplication of Complex Numbers	17
8	The Effect of Stretching a Picture Function by a Factor s in the X Direction	20
9	A Third Level GBT Aggregate	23
10	The Geometry of the GBT Hexagonal Grid	23
11	Values of f^2 for the 28 Rotationally Distinct Classes of Binary Picture Functions on First Level GBT Aggregates	31
12	Sample Texture 1. From a Photograph of Loose Woven Fabric	37
13	Sampl Texture 2. From a Photograph of Pleated Fabric	38
14	Sample Texture 3. From a Photograph of Herringbone Wool Fabric	39
15	Sample Texture 4. From a Photograph of Nubby Wool Fabric	40
16	Sample Texture 5. From a Photograph of Straw	41
17	A Portion of a Raster Scanned Diagram. The Descriptors for the Fourth Level Aggregate A are given in Table 9.	43
18	Based on Descriptors, a Line Segment is computed for the Aggregate A	44
19	A Portion of a Raster Scanned Diagram. The Descriptors for the Fourth Level Aggregate B are given in Table 10	45
20	The Aggregate B and its Third Level Subordinates	46
21	Based on Descriptors, Line Segments are computed for the Subordinates of B	47

LIST OF TABLES

1	Basic Statistics for an n-gon of Radius R	9
2	Statistics for the Rectangle, Ellipse, and Rhombus	12
3	Values of f and f^2 for the regular n-gon as n goes from 3 to 30.	15
4	Formulas for f^2 for Various Types of Picture Functions	16
5	The Correspondence between Addresses in the Central First Level Aggregate and Complex Numbers	24
6	Values of f and f^2 for a picture function which is one within a level n GBT aggregate and zero elsewhere	30
7	Mean Vectors and Standard Deviations for Sample Textures	36
8	Mahalanobis Distance. A row contains the average, maximum, and minimum Mahalanobis distance to the mean of a texture as computed for all samples of the texture represented by the column	36
9	Descriptors for Level Four Aggregate A in Figure 17	42
10	Descriptors for Level Four Aggregate B in Figure 19 and for its Subordinates shown in Figure 20	48
11	Questions and Answers about Research on Texture	51
12	Questions and Answers about Research on Shape Descriptors	52

1.0 INTRODUCTION

Current generation automated systems have only a limited capability for interacting with their environment. This is due, in large part, to the fact they cannot understand their environment. The lack of understanding is not caused by a lack of data. A variety of sensors are available to capture information in digital form. The problem is to understand this information.

The goal of this research was to develop new tools to aid in the automated analysis of two dimensional arrays of pixels. Such arrays can be produced by optical, infrared, or radar sensors. Understanding pixel arrays by examining individual pixels is difficult because each pixel contains only a shred of evidence about the entity to which it belongs. Looking at all the pixels together usually presents too complicated a mass of information to be easily understood. For these reasons our theoretical development and experimentation have all been based on the GBT method of successively decomposing two dimensional space. By using GBT cells of varying sizes, we are able to find areas of the image that contain enough pixels to represent something significant but not so many as to defy comprehension.

The tools developed during this research are the function f^2 that provides information about shape and the vector T that provides a local quantification of image texture. High values of f^2 over a region indicate that the region contains a solid, well compacted object, e.g, a rectangle or ellipse. Low values indicate the image is fragmented. In the context of GBT, f^2 tells whether a region can be understood as a whole or whether it must be decomposed.

The T vector allows the computation of the texture of any region as a convex sum of the textures of its constituent parts. Preliminary results have shown that what the T vector measures corresponds closely to the human perception of visual texture.

The result of this research is that two valuable tools for automated image understanding have been developed and proven valuable in limited experimentation. More work is required to see what further tools can be developed and to exploit the full power of the tools we have now.

2.0 STATEMENT OF THE PROBLEM

One of the major weakness of today's automated systems is their lack of ability to convert analog data into internally usable form. This applies particularly to two dimensional images. There are important military applications for understanding the content of imagery from a variety of sources: images from camera carrying platforms, radar images, and infrared images. Current automated techniques, however, have only primitive capabilities.

The capability for converting analog images to digital form as an array of pixels exists and is relatively mature. Pixel digitization is even accomplished in some systems at the time of image capture. The question is: what do all those pixels mean? Taken individually, each pixel can contribute only the tiniest understanding of the overall image content. Yet the totality of pixels, taken together, usually are too complicated for automated understanding. What is required is a method of examining regions of various sizes and shapes so that entities that are both significant and simple can be isolated from the rest of the image and analyzed.

The research reported here lays the foundation for performing this process. It uses the Generalized Balanced Ternary (GBT) method of structuring space into a cellular hierarchy as the basis for analyzing the behavior of picture functions. It shows how to use shape descriptors, both classical and original, to understand the nature and complexity of the picture function at cells throughout the GBT hierarchy. It further gives techniques for piecing together the rigidly structured GBT cells to form regions of arbitrary shape.

Besides its innovative use of the GBT structure in analyzing imagery, two new tools have been derived during this research: a function called f^2 which gives information about the shape of an image within a region and a quantification of regional texture as a vector.

These new tools and the easy ways they interact within the GBT hierarchy provide a significant advance along the path toward automated image understanding.

3.0 SCOPE

The research performed under this contract was done from April 1, 1982 to October 1, 1982 by Dr. Dean Lucas, Dr. Laurie Gibson, Mr. Tom Bundy, Mr. Jay Elston, and Mr. Terry Elmore. Past research in the mathematics of Generalized Balanced Ternary and its application to processing binary images was extended by this work. In addition, commercial and experimental software systems previously developed at Interactive Systems Corporation were used in the project. The objectives of this research were as follows:

1. Demonstrate the capability of the Generalized Balanced Ternary approach to extract from pixel data the features and regions of common texture in an aerial photograph.

2. Derive aggregate measures for the representation of cartographic information to be stored in a data file used in feature recognition and change detection.

The research proceeded in these steps:

1. Convert existing software for scanning and processing binary images to handle the scanning of photographs. Develop histograming, image enhancement, display, and hard copy functions.

2. Define texture.

3. Build software for computing texture and test on photographs.

4. Build software for the statistical analysis of texture (compute parameters of a distribution and Mahalanobis distance).

5. Analyze the texture definition and its ability to distinguish intuitive textures.

6. Define aggregate descriptors.

7. Modify existing software to compute these descriptor functions for the scanned images.

8. Test the descriptors on actual images and refine definitions. Select those that seem most useful.

During the course of this project the following papers were presented which announced preliminary results of the research.

1. "Spatial Data Processing using Generalized Balanced Ternary" presented by Dean Lucas at the IEEE Computer Society Conference on Pattern Recognition and Image Processing, June 1982, Las Vegas, Nevada.

2. "Automated Data Capture" presented by Laurie Gibson at the Symposium on Automation Technology for Management and Productivity Advancement through CAD/CAM and Data Handling, November, 1982, Naval Postgraduate School, Monterey, California.

4.0 DESCRIPTION OF THE TECHNOLOGY

4.1 Picture Functions

The goal of this research was to develop tools to enable a computer to understand the content of a picture. In order to understand the progress of the research and the nature of the tools that were developed, it is necessary to start at the beginning, with the basic notion of a picture function.

For our purposes, a picture function, p, is a real valued function defined on the plane. In what follows we will address the points in the plane as complex numbers, i.e., numbers of the form $x + yi$ where x and y are real numbers and $i^2 = -1$. In practice, the values that a picture function can take on are often restricted. Picture functions which take on the values 0 and 1 only are called binary picture functions. We will use the term gray level when speaking of a picture function that takes on integer values from 0 to 255 inclusive. Figure 1 is an illustration of a binary picture function. The illustration is meant to be interpreted as follows: at every point at which ink is present on the paper the picture function takes on the value 1, where ink is not present it takes on the value 0.

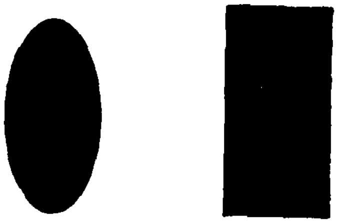


Figure 1. An Illustration of a Binary Picture Function.

Picture functions can be added together, multiplied together, and multiplied by scalars to yield new picture functions. In what follows we will take advantage of these algebraic properties of picture functions to try to express complicated picture functions as sums or differences of simple picture functions. For instance, the picture function illustrated in Figure 1 might be expressed as the sum of two more easily understandable picture functions: one an ellipse, the other a rectangle.

4.2 Classical Techniques

We are interested in developing tools for determining the content of pictures. Our techniques are based on some relatively well known analysis functions. In this section we restate these functions to remind the reader of their existence and introduce our notation. In what follows, p is a picture function on the complex plane, $z = x + yi$ is a variable complex number, c is a fixed complex number, and R is a region of the plane.

For $q = 0, 1, \cdots$, and $m = 0, 1, \cdots$, the qth moment of p^m about c in the region R is:

$$M(q,c,p^m,R) = \int_{z \in R} (z-c)^q p(z)^m dz$$

Let us illustrate and name some of the simpler of these moments. (The question of the existence of these integrals is not addressed since it has no practical consequences.)

$$M(0,c,p^0,R) = \int_{z \in R} dz = \text{the area of R.}$$

$$M(0,c,p,R) = \int_{z \in R} p(z)dz = \text{the weight of p on R.}$$

$$M(1,c,p,R) = \int_{z \in R} (z-c)\ p(z)dz = \text{the first moment about c of p on R.}$$

$$M(2,c,p,R) = \int_{z \in R} (z-c)^2\ p(z)dz = \text{the second moment about c of p on R.}$$

There are several other classical functions that are related to or derived from moments. We define and name those functions that we will need later.

$$\mu(p,R) = M(0,c,p^1,R)/M(0,c,p^0,R) = \text{the mean of p on R.}$$

$$Var(p,R) = M(0,c,p^2,R)/M(0,c,p^0,R) - \mu(p,R)^2 = \text{the variance of p on R.}$$

$$S(c,p,R) = \int_{z \in R} |z-c|^2\ p(z)dz = \text{the scalar moment about c of p on R.}$$

$$c^*(p,R) = c + \frac{M(1,c,p,R)}{M(0,c,p,R)} = \int_{z \in R} z\ p(z)dz / \int_{z \in R} p(z)dz$$

= the centroid of p on R (note that the centroid is independent of the fixed point c).

For many applications we will wish to use the centroid $c^*(p,R)$ as the fixed point c about which we compute moments. Below we state several classical results *concerning moments* about centroids of picture functions.

$$\lambda_1(p,R) = \tfrac{1}{2}(S(c^*,p,R) + |M(2,c^*,p,R)|)/M(0,c^*,p,R)$$

= the <u>primary eigenvalue</u> of p on R.

$$\lambda_2(p,R) = \tfrac{1}{2}(S(c^*,p,R) - |M(2,c^*,p,R)|)/M(0,c^*,p,R)$$

= the <u>secondary eigenvalue</u> of p on R.

$M(1,c^*,p,R)$ = 0. In words, the first moment about the centroid is zero.

$Ax(p,R)$ = ½(angle of $M(2,c^*,p,R)$

= the <u>angle of the principal axis of inertia</u> of p on R.

It may happen that we have computed moments and the scalar moment about a point c which is not the centroid (the origin for instance). Since the second moment and the scalar moment about the centroid are needed for computing eigenvalues, it is handy to have a means of obtaining them directly. The formulas below provide such a means:

$$M(2,c^*,p,R) = M(2,c,p,R) - M(1,c,p,R)^2/M(0,c,p,R)$$

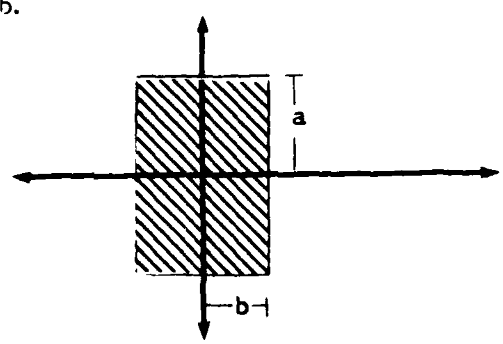
$$S(c^*,p,R) = S(c,p,R) - M(1,c,p,R)\,\bar{M}(1,c,p,R)/M(0,c,p,R),$$

where $\bar{M}(1,c,p,R)$ *is the complex conjugate of* $M(1,c,p,R)$

4.3 <u>Examples</u>

In order to show the utility of these classical analysis functions and to lead into the discussion of other functions, it is worth looking at some simple examples.

Let p be the binary picture function representing a rectangle of half height a and half width b centered at the origin. Figure 2 illustrates this picture function. Some relevant statistics for this picture function have been computed and are shown beside the illustration. Figures 3 and 4 show the comparable illustration and statistics for an ellipse and a rhombus both centered at the origin with semiaxes a and b.



weight = 4ab

first moment = 0

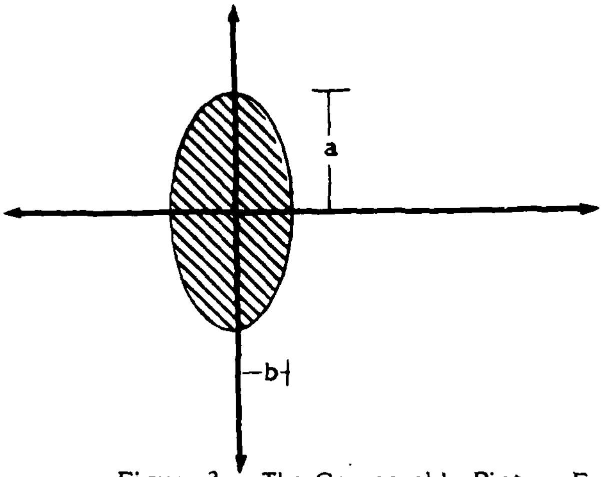
second moment = $\frac{4}{3}(b^3a - a^3b)$

scalar moment = $\frac{4}{3}(b^3a + a^3b)$

$$\lambda_1 = \begin{cases} a^2/3 & \text{if } a \geq b \\ b^2/3 & \text{if } b > a \end{cases}$$

$$\lambda_2 = \begin{cases} b^2/3 & \text{if } a \geq b \\ a^2/3 & \text{if } b > a \end{cases}$$

Figure 2. A Binary Picture Function of a Rectangle centered at the Origin together with significant Statistics.



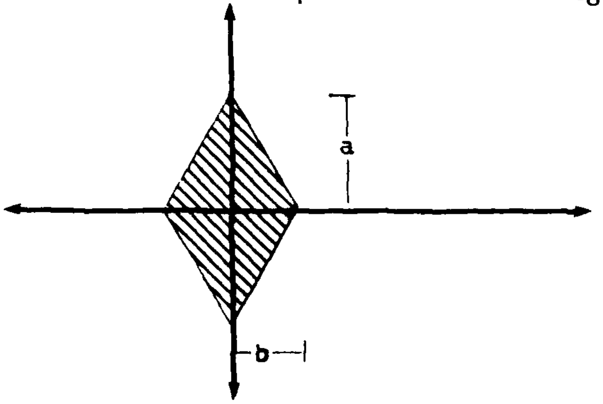
weight = πab
first moment = 0
second moment = $\frac{\pi}{4}(b^3a - a^3b)$

scalar moment = $\frac{\pi}{4}(b^3a + a^3b)$

$$\lambda_1 = \begin{cases} a^2/4 & \text{if } a \geq b \\ b^2/4 & \text{if } b > a \end{cases}$$

$$\lambda_2 = \begin{cases} b^2/4 & \text{if } a \geq b \\ a^2/4 & \text{if } b > a \end{cases}$$

Figure 3. The Comparable Picture Function and Statistics for an Ellipse centered at the Origin with Semiaxes a and b.



weight = 2ab
first moment = 0
second moment = $\frac{1}{3}(b^3a - a^3b)$

scalar moment = $\frac{1}{3}(b^3a + a^3b)$

$$\lambda_1 = \begin{cases} a^2/6 & \text{if } a \geq b \\ b^2/6 & \text{if } b > a \end{cases}$$

$$\lambda_2 = \begin{cases} b^2/6 & \text{if } a \geq b \\ a^2/6 & \text{if } b > a \end{cases}$$

Figure 4. The Comparable Picture Function and Statistics for a Rhombus centered at the Origin with Semiaxes a and b.

To get a feel for the computation involved and to provide one more useful example, let us explicitly derive the moments and scalar moment for a regular n-gon of outer radius R. Specifically, for each integer n greater than two, we will construct a regular n-gon centered at the origin from n isosceles triangles similar to the triangle, T, shown in Figure 5. The n-gon is built as a union of n of these triangles each successive triangle being rotated through an angle of $2\pi/n$ from the position of its predecessor. Thus, the set, S, of points in the n-gon is:

(1) $$S = \{ze^{j2\pi i/n} \mid z\epsilon T,\ j = 0, 1, \cdot\cdot\cdot, n\text{-}1\}$$

where T is the triangle shown in Figure 5 and $i^2 = -1$. The picture function, p, which describes this n-gon takes the value 1 on all points in S and 0 elsewhere.

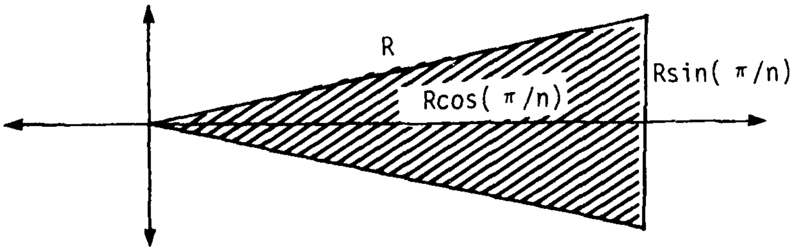


Figure 5. The Basic Triangle, T, from which an N-gon is constructed.

The qth moment of p^m about the origin is computed as follows:

$$M(q,0,p^m,C) = \int_{z\epsilon C} z^q\, p(z)^m dz \text{ , by definition}$$

$$= \int_{z\epsilon C} z^q\, p(z) dz, \text{ since } p(z)^m = p(z)$$

$$= \int_{z\epsilon S} z^q\, dz, \text{ since } p(z) = 1 \text{ within the n-gon, 0 elsewhere}$$

$$= \sum_{j=0}^{n-1} \int_{z\epsilon T} z^q\, e^{jq2\pi i/n}\, dz, \text{ from (1)}$$

(2) $$= \sum_{j=0}^{n-1} e^{jq2\pi i/n} \int_{z\epsilon T} z^q\, dz.$$

Let us examine the first portion of this expression. It is a geometric series in $e^{q2\pi i/n}$. If q is a multiple of n then $e^{q2\pi i/n} = 1$ and the series sums to n. If q is not a multiple of n then $e^{q2\pi i/n} \neq 1$ and the following holds:

$$\sum_{j=0}^{n-1} e^{jq2\pi i/n} = \frac{1 - e^{q2\pi i}}{1 - e^{q2\pi i/n}} = 0 \text{ since } e^{q2\pi i} = 1 \text{ and the denominator is not zero.}$$

Now let us compute the integral which is the second part of (2).

$$\int_{z\epsilon T} z^q dz = \int_0^{R\cos(\pi/n)} \int_{-x\tan(\pi/n)}^{x\tan(\pi/n)} (x + yi)^q\, dy\, dx, \text{ since these limits describe T}$$

$$= \frac{(1 + i\tan(\pi/n))^{q+1} - (1 - i\tan(\pi/n))^{q+1}}{i(q + 1)(q + 2)} R^{q+2} \cos^{q+2}(\pi/n),$$

by standard integration.

From these two considerations we can describe all moments of the n-gon of outer radius R described above:

(3) $$M(q,0,p^m,C) = \begin{cases} 0, \text{ if q is not a multiple of n} \\ nR^{q+2}\cos(\pi/n)^{q+2}\dfrac{(1 + i\tan(\pi/n))^{q+1} - (1 - i\tan(\pi/n))^{q+1}}{i(q + 1)(q + 2)} \\ \text{otherwise.} \end{cases}$$

In particular the first and second moments of the n-gon are always zero since n must be greater than two. The scalar moment is computed similarly:

$$
\begin{aligned}
S(0,p,C) &= \int_{z \epsilon C} |z|^2 \, p(z)dz, \text{ by definition} \\
&= \int_{z \epsilon S} |z|^2 \, dz, \text{ since } p(z) = 1 \text{ in } S \text{ and } p(z) = 0 \text{ elsewhere} \\
&= \sum_{j=0}^{n-1} \int_{z \epsilon T} |ze^{jq2\pi i/n}|^2 \, dz, \text{ from (1)} \\
&= \sum_{j=0}^{n-1} \int_{z \epsilon T} |z|^2 dz, \text{ since } |e^{jqz\pi i/n}| = 1 \\
&= n \int_0^{R\cos(\pi/n)} \int_{-x\tan(\pi/n)}^{x\tan(\pi/n)} x^2 + y^2 \, dy \, dx
\end{aligned}
$$

$$
= \tfrac{1}{2} n(\tan(\pi/n) + \tan^3(\pi/n)/3)R^4 \cos^4(\pi/n) \tag{4}
$$

Since the first moment of the n-gon is zero its centroid is at the origin and the moments we have just computed are, in fact, moments about the centroid. By examining the formulas for eigenvalues given above, we can see that the effect of the second moment being zero is that both eigenvalues are equal and are explicitly computed as follows:

$$
\lambda_1(p,C) = \lambda_2(p,C) = \frac{S(0,p,C)}{2\,M(0,0,p,C)} \tag{5}
$$

$$
= \tfrac{1}{4}(1 + \tan^2(\pi/n)/3)R^2 \cos^2(\pi/n), \text{ by (3) and (4)}
$$

The results of these computations are summarized in Table 1.

weight	$= n \tan(\pi/n) R^2 \cos^2(\pi/n) = nR^2 \sin(\pi/n) \cos(\pi/n)$
first moment	$= 0$
second moment	$= 0$
scalar moment	$= \tfrac{1}{2} n(\tan(\pi/n) + \tan^3(\pi/n)/3) R^4 \cos^4(\pi/n)$
$\lambda_1 = \lambda_2$	$= \tfrac{1}{4} (1 + \tan^2(\pi/n)/3) R^2 \cos^2(\pi/n)$

Table 1. Basic Statistics for an N-gon of Radius R.

As a final example, we will compute the usual classical statistics for the binary picture function illustrated in Figure 6. The picture function takes the value 1 within each of the n rectangles shown and is 0 elsewhere. The computational steps will be somewhat more abbreviated than in the previous example, leaving the reader to fill in the details.

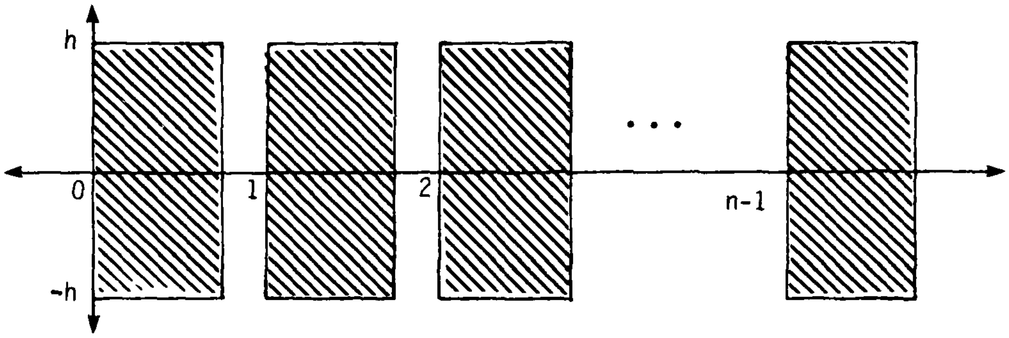


Figure 6. N identical rectangles aligned and equally spaced. Rectangles are of height 2h and width w ($0 \leq w \leq 1$).

$$M(q,0,p,C) = \int_{z \in C} z^q \, p(z) dz$$

$$= \sum_{j=0}^{n-1} \int_j^{j+w} \int_{-h}^{h} (x + yi)^q \, dy \, dx$$

$$= \frac{1}{i(q+1)(q+2)} \sum_{j=0}^{n-1} \left(\sum_{k=1}^{q+1} \binom{q+z}{k} w^k ((j+hi)^{q+2-k} - (j-hi)^{q+2-k}) \right)$$

In particular, by substituting q = 0, 1, and 2, we get:

$$\text{weight} = \frac{1}{2i} \sum_{j=0}^{n-1} 4hiw = 2nhw$$

$$\text{first moment about } 0 = \frac{1}{6i} \sum_{j=0}^{n-1} (12jhiw + 6hiw^2)$$

$$= nhw(n + w - 1)$$

$$\text{second moment about } 0 = \frac{1}{12i} \sum_{j=0}^{n-1} (24j^2hiw - 8h^3iw + 24jhiw^2 + 8hiw^3)$$

$$= \frac{nhw}{3} (2n^2 - 3n + 1 - 2h^2 + 3nw - 3w + 2w^2)$$

Computing the scalar moment about the origin goes as follows:

$$S(0,p,C) = \int_{z \epsilon C} | z |^2 p(z)dz$$

$$= \sum_{j=0}^{n-1} \int_j^{j+w} \int_{-h}^{h} x^2 + y^2 \, dy \, dx$$

$$= \frac{nhw}{3} (2n^2 - 3n + 1 + 2h^2 + 3nw - 3w + 2w^2)$$

From the weight and first moment we obtain the centroid:

$$c^*(p,C) = \frac{nhw(n + w - 1)}{2nhw} = \frac{n + w - 1}{2}$$

Then the second moment and scalar moment about the centroid are computed from the formulas provided above:

$$M(2,c^*,p,C) = M(2,0,p,C) - M(1,0,p,C)^2/M(0,0,p,C)$$

$$= \frac{nhw}{6} (n^2 + w^2 - 1 - 4h^2)$$

$$S(c^*,p,C) = S(0,p,C) - M(1,0,p,C)\bar{M}(1,0,p,C)/M(0,0,p,C)$$

$$= \frac{nhw}{6} (n^2 + w^2 - 1 + 4h^2)$$

In this example $M(2,c^*,p,C)$ is a real number which may be positive or negative depending on the size of h. Therefore, $| M(2,c^*,p,C) | = \pm M(2,c^*,p,C)$. It follows that the eigenvalues λ_1 and λ_2 are the maximum and minimum, respectively, of the following two expressions:

$$\left.\begin{matrix}\lambda_1 \\ \lambda_2\end{matrix}\right\} = \left\{\begin{matrix}\max \\ \min\end{matrix}\right. (\tfrac{1}{2}(S(c^*,p,C) \pm M(2,c^*,p,C))/M(0,c^*,p,C)$$

$$= \left\{\begin{matrix}\max \\ \min\end{matrix}\right. (\frac{nhw}{12} (n^2 + w^2 - 1 + 4h^2$$

$$\pm (n^2 + w^2 - 1 - 4h^2))/2nhw)$$

$$= \left\{\begin{matrix}\max \\ \min\end{matrix}\right. (\frac{n^2 + w^2 - 1}{12}, \frac{h^2}{3}).$$

4.4 f and f^2

We now wish to use some of the classical pattern descriptors that we have been discussing to derive additional information about shape. The first three examples in the previous section provide the motivation. These were: a rectangle, an ellipse, and a rhombus, each with semiaxes of length a and b. In each case the weight was a multiple of the products of the lengths of the semiaxes by a factor that we will call s. In each case the principal and secondary eigenvalues were, respectively, the squares of the semimajor and semiminor axes divided by a factor that we will call t:

$$\begin{aligned} \text{weight} &= sab \\ \lambda_1 &= \max \{a^2/t, b^2/t\} \\ \lambda_2 &= \min \{a^2/t, b^2/t\} \end{aligned}$$

For these three examples s and t take on the values given in Table 2. Although the relations described above are not true for all picture functions (a and b are not always well defined), their essence can be exploited in deriving a general shape description. Notice that:

$$\text{weight}^2 = s^2a^2b^2$$

and $$\lambda_1\lambda_2 - a^2b^2/t^2$$

thus, $$\frac{\text{weight}^2}{\lambda_1\lambda_2} = s^2t^2.$$

	Weight	$\{\lambda_1,\lambda_2\}$	s	t	f	f^2
Rectangle	4ab	$\{a^2/3,b^2/3\}$	4	3	6	36
Ellipse	πab	$\{ a^2/4, b^2/4 \}$	π	4	2π	$4\pi^2$
Rhombus	2ab	$\{ a^2/6, b^2/6 \}$	2	6	6	36

Table 2. Statistics for the Rectangle, Ellipse, and Rhombus.

The left-hand side of this expression contains terms that are computable fcr any picture function while the right-hand combines two unknowns which individually contain information about the shape of certain binary picture functions. With the hope that the left-hand side may be of use as a shape descriptor for arbitrary picture functions, we make the following definition:

$$f(p,R) = \left(\frac{M(0,c^*,p,R)^2}{4\lambda_1(p,R)\lambda_2(p,R)}\right)^{½}$$

and, therefore, $$f(p,R)^2 = \frac{M(0,c^*,p,R)^2}{4\lambda_1(p,R)\lambda_2(p,R)}$$

The four in the denominator is inserted to make some other expressions for f^2 less cluttered. For instance, by substituting the formulas for λ_1 and λ_2 in terms of moments into the above definition for f^2, one obtains:

$$f^2(p,R) = \frac{M(0,c^*,p,R)^4}{S(c^*,p,R)^2 - |M(2,c^*,p,R)|^2}$$

In what follows let us consider the picture function, p, and the region, R, to be implicit and suppress them from the notation. Thus, the above expression becomes:

$$f^2 = \frac{M(0,c^*)^4}{S(c^*)^2 - |M(2,c^*)|^2}$$

To express the f^2 function in terms of moments about an arbitrary point, c, we make the following observations:

$$M(0,c^*)^4 = M(0,c)^4 \text{ for any c.}$$

$$S(c^*)^2 = (S(c) - \frac{M(1,c)\,\bar{M}(1,c)}{M(0,c)})^2$$

$$= S(c)^2 - 2S(c)\frac{M(1,c)\,\bar{M}(1,c)}{M(0,c)} + \frac{M(1,c)^2\,\bar{M}(1,c)^2}{M(0,c)^2}$$

$$|M(2,c^*)|^2 = (M(2,c) - \frac{M(1,c)^2}{M(0,c)})(\bar{M}(2,c) - \frac{\bar{M}(1,c)^2}{M(0,c)})$$

$$= M(2,c)\bar{M}(2,c) - M(2,c)\frac{\bar{M}(1,c)^2}{M(0,c)} - \bar{M}(2,c)\frac{M(1,c)^2}{M(0,c)} + \frac{M(1,c)^2\bar{M}(1,c)^2}{M(0,c)^2}$$

therefore

$$S(c^*)^2 - |M(2,c^*)|^2 = S(c)^2 - |M(2,c)|^2 + \frac{1}{M(0,c)}(M(2,c)\bar{M}(1,c)^2 + \bar{M}(2,c)M(1,c)^2 - 2S(c)|M(1,c)|^2)$$

and

$$f^2 = \frac{M(0,c)^4}{S(c)^2 - |M(2,c)|^2 + (M(2,c)\bar{M}(1,c)^2 + \bar{M}(2,c)M(1,c)^2 - 2S(c)|M(1,c)|^2)/M(0,c)}$$

Table 2 contains values of f and f^2 for the rectangle, ellipse, and rhombus. Notice that the values are independent of the lengths of the semiaxes, a and b, and are identical for the rectangle and rhombus. This is no accident as we shall see when we discuss invariants. Before going on let us see what can be learned by applying the f^2 function to our other two examples: the regular n-gon and the aligned rectangles. For the n-gon, we use the definition of f^2 in terms of moments about the centroid and take advantage of the fact that the second moment about the centroid is zero:

$$
\begin{aligned}
f^2 &= \frac{M(0,c^*)^4}{S(c^*)^2 - |M(2,c^*)|^2} \\
&= \frac{(n\ \tan(\pi/n)\ R^2\ \cos^2(\pi/n))^4}{(\tfrac{1}{2}n(\tan(\pi/n) + \tan^3(\pi/n)/3)\ R^4 \cos^4(\pi/2))^2} \\
&= \frac{4n^2\ \tan^2(\pi/n)}{(1 + \tan^2(\pi/n)/3)^2}
\end{aligned}
$$

so,
$$f = \frac{2n\ \tan(\pi/n)}{1 + \tan^2(\pi/n)/3}$$

Table 3 gives the values of f and f^2 for the regular n-gon as n goes from 3 to 30. Some aspects of the table are worth examining. For an equilateral triangle (3-gon), f^2 is exactly 27. For a square, f^2 is 36. This latter result was amply clear from Table 2 since a square is a special case of both a rectangle and a rhombus. For a regular hexagon, f^2 is 38.88 ($2^2 3^5/5^2$ exactly). This result will be of interest later. As for the rest of the table, it appears that the values of f are increasing and converging. By examining the derivative of f and the limit of f as n goes to infinity one can prove that f is an increasing function of n that converges to 2π. That f converges to 2π for an n-gon was to be expected from Table 2 since f on an ellipse, hence a circle, was 2π.

n	f	f^2
3	5.1962	27.0000
4	6.0000	36.0000
5	6.1783	38.1716
6	6.2354	38.8800
7	6.2583	39.1657
8	6.2689	39.2990
9	6.2744	39.3681
10	6.2775	39.4068
11	6.2793	39.4299
12	6.2805	39.4444
13	6.2812	39.4538
14	6.2817	39.4602
15	6.2821	39.4646
16	6.2823	39.4678
17	6.2825	39.4701
18	6.2827	39.4718
19	6.2828	39.4731
20	6.2828	39.4741
21	6.2829	39.4749
22	6.2830	39.4755
23	6.2830	39.4760
24	6.2830	39.4763
25	6.2830	39.4767
26	6.2831	39.4769
27	6.2831	39.4771
28	6.2831	39.4773
29	6.2831	39.4775
30	6.2831	39.4776

Table 3. Values of f and f^2 for the regular n-gon as n goes from 3 to 30.

Let us see how f^2 behaves on our last example from the previous section, the alignment of n identical rectangles shown in Figure 6. Using the formula for f^2 in terms of moments about the centroid,

$$
\begin{aligned}
f^2 &= \frac{36(2nhw)^4}{(nhw)^2((n^2 + w^2 - 1 + 4h^2)^2 - (n^2 + w^2 - 1 - 4h^2)^2)} \\
&= \frac{36 \text{ x } 16(nhw)^2}{4(4n^2h^2 + 4w^2h^2 - 4h^2)} \\
&= \frac{36n^2w^2}{n^2 + w^2 + 1}.
\end{aligned}
$$

Recall that n is a positive integer giving the number of rectangles and w is a number between 0 and 1 giving the width of each rectangle. When n or w is 1, f^2 takes the value 36. For the former case there is just one rectangle and in the latter the n rectangles are merged into one long rectangle. Therefore, the value 36 is consistent with our earlier observations on rectangles.

f^2 decreases as w decreases, going to zero as w goes to zero. This property gives some insight into the behavior of f^2: low values of f^2 on binary picture functions indicate relatively small dark areas separated by relatively large spaces. f^2 decreases as n increases, converging to $36w^2$ as n goes to infinity. This means that whenever n is large, f/6 is a good estimator for the proportion of darkened area along a line of identical rectangles.

This example is of more than academic interest since lines of identical (or almost identical) rectangles occur frequently in applications. When h is small compared with w, a line of rectangles looks like a dashed line. When h is very large compared with w, a line of rectangles looks like a family of parallel lines. The value of f^2, however, is independent of h. This is no accident as will be evident below.

Table 4 lists the formulas we have derived for f^2 based on our examples together with some other formulas that the reader is encouraged to derive independently.

Shape Described by Binary Picture Function	f^2
Rectangle	36
Ellipse	$4\pi^2$ (=39.4784)
Rhombus	36
Regular n-gon	$\left(\frac{2n\tan(\pi/n)}{1 + \tan^2(\pi/n)/3}\right)^2$
Line of n rectangles of width w $0 \le w \le 1$)	$\frac{36n^2w^2}{n^2 + w^2 - 1}$
Semicircle	$\frac{9\pi^4}{9\pi^2 - 64}$ (= 35.3124)
Annulus of inner and outer radii r and R	$4\pi^2 \left(\frac{R^2 - r^2}{R^2 + r^2}\right)^2$

Table 4. Formulas for f^2 for Various Types of Picture Functions.

4.5 Properties of f and f^2

Throughout this section we will assume that the region, R, within which we are examining a picture function, is the entire complex plane, C. This allows a slight notational simplification. For instance, we will write M(q,c,p,C) as M(q,c,p).

We are concerned in this section with what happens to our various measures of picture properties, particularly f and f^2, when the picture undergoes certain simple changes. The first change we will deal with is translation, rotation, and scaling of a picture. Specifically, if we have an existing picture function, p, we will construct a new picture function, p^*, which takes on the same value at the point v(z + w) as p takes on at the point z:

$$p^*(v(z + w)) = p(z).$$

Here v and w are fixed complex numbers. Adding w to z has the effect of translating the picture by the vector w. Multiplying z + w by v rotates z + w about the origin by the angle of v and scales z + w in magnitude by the factor $|v|$. These operations are illustrated in Figure 7.

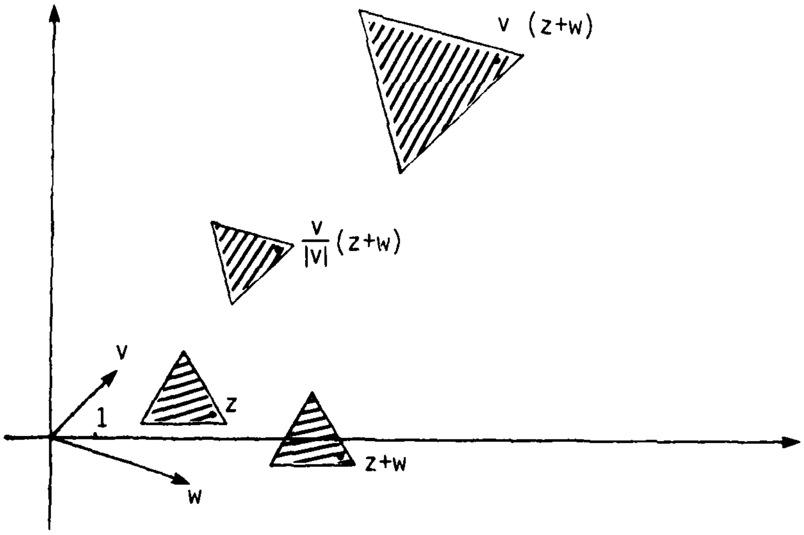


Figure 7. How Translation, Rotation, and Scaling are accomplished by Addition and Multiplication of Complex Numbers.

Let $z^* = v(z + w)$. Then the following relations hold:

$$p^*(z^*) = p^*(v(z + w)) = p(z), \text{ by definition}$$

$$(z^* - c)^q = (vz + vw - vc/v)^q = v^q(z - (c/v - w))^q$$

$$dz^* = |v|^2 dz, \text{ the Jacobian of the scaling transformation is } |v|^2$$

so, $$M(q,c,p^*) = \int_{z^* \in C} (z^* - c)^q p^*(z^*) d(z^*)$$

$$= v^q \, |v|^2 \int_{z \in C} (z - (c/v - w))^q p(z) dz$$

(6) $$= v^q \, |v|^2 \, M(q,\ c/v - w,\ p)$$

or, alternatively, $$= v^q \, |v|^2 \int_{z \in C} \sum_{k=0}^{q} \binom{q}{k} z^k (w - c/v)^{q-k} p(z) dz$$

(7) $$= v^q \, |v|^2 \sum_{k=0}^{q} \binom{q}{k} (w - c/v)^{q-k} M(k,0,p)$$

Similarly, $$S(c,p^*) = \int_{z^* \in C} |z^* - c|^2 \, p^*(z^*) dz^*$$

(8) $$= |v|^4 \, S(c/v - w,\ p)$$

Letting q be 0 and 1 in (7) we find:

$$M(0,c,p^*) = |v|^2 M(0,0,p)$$

$$M(1,c,p^*) = v \, |v|^2 \, ((w - c/v)M(0,0,p) + M(1,0,p))$$

Therefore, $$c^*(p^*) = c + \frac{M(1,c,p^*)}{M(0,c,p^*)},$$ the definition of the centroid of p^* on C

$$= c + v(w - c/v + \frac{M(1,0,p)}{M(0,0,p)})$$

$$= v(c^*(p) + w)$$

so that $$c^*(p) = \frac{c^*(p^*)}{v} - w$$

This last relation, when plugged into (6) and (8), makes the comparison of moments about centroids easy:

$$M(q,c^*(p^*),p^*) = v^q \, |v|^2 \, M(q,c^*(p),p)$$

$$S(c^*(p^*),p^*) = |v|^4 \, S(c^*(p),p)$$

The relations among eigenvalues and f^2 for p^* and p then follow immediately:

$$\lambda_1(p^*) = ½(S(c^*(p^*),p^*) + |M(2,c^*(p^*),p^*)|)/M(0,c^*(p^*),p^*)$$

$$= |v|^2 \lambda_1(p)$$

similarly, $\lambda_2(p^*) = |v|^2 \lambda_2(p)$

and $$f^2(p^*) = \frac{M(0,c,p^*)^2}{4\lambda_1(p^*)\lambda_2(p^*)}$$

$$= f^2(p)$$

This last result is important in understanding the utility of the f function: the value of f does not change when a picture is translated, rotated, and scaled.

The f function has one other useful property that we will now proceed to demonstrate: the value of f does not change when a picture is stretched along any single axis. This is different from the scaling operation treated above since multiplication by a complex number scales in two dimensions. Here we are referring to scaling in one arbitrary dimension while leaving the orthogonal dimension unaltered. This type of operation, for instance, will convert a circle into an ellipse.

We have already seen that f(p) is unchanged by translation and rotation of the picture function p. Therefore, we can assume that the picture function that we will be stretching has its centroid at the origin and is to be stretched in the direction of the X axis. No generality is lost since any picture function can be translated until its centroid is at the origin and rotated until the axis to be stretched lies along the X axis.

With these caveats we construct from our picture function, p, a new picture function, p^*, which takes on the same value at the point sx + yi as p takes on at x + yi. Thus

$$p^*(sx + yi) = p(x + yi)$$

Here s is a real number corresponding to the amount of stretch along the X axis. The effect of this operation is illustrated in Figure 8. Since we will be dealing for the first time with integrals in the real variables x and y it is convenient to introduce some new notation:

$$I(m,n) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^m y^n p(x + yi)dxdy.$$

Letting $z^* = sx + yi$, the following holds:

$$p^*(z^*) = p^*(sx + yi) = p(x + yi) = p(z)$$

$$z^{*q} = (sx + yi)^q = \sum_{k=0}^{q} \binom{q}{k} s^k i^{q-k} x^k y^{q-k}$$

$dz^* = sdxdy$, s is the Jacobian of the stretching transformation

so, $$M(q,0,p^*) = \int_{z^* \epsilon C} z^{*q}\, p^*(z^*)\, dz^*$$

$$= s \sum_{k=0}^{q} \binom{q}{k} s^k i^{q-k} \iint x^k y^{q-k} p(x+yi)\,dxdy$$

(9) $$= \sum_{k=0}^{q} \binom{q}{k} s^{k+1} i^{q-k} I(k,q-k)$$

Letting q be 0 and 1 we find:

$$M(0,0,p^*) = sI(0,0)$$

$$= sM(0,0,p)$$

$$M(1,0,p^*) = siI(0,1) + s^2 I(1,0)$$

so that $$c^*(p^*) = \frac{M(1,0,p^*)}{M(0,0,p^*)} = s\frac{I(1,0)}{I(0,0)} + \frac{I(0,1)}{I(0,0)}i$$

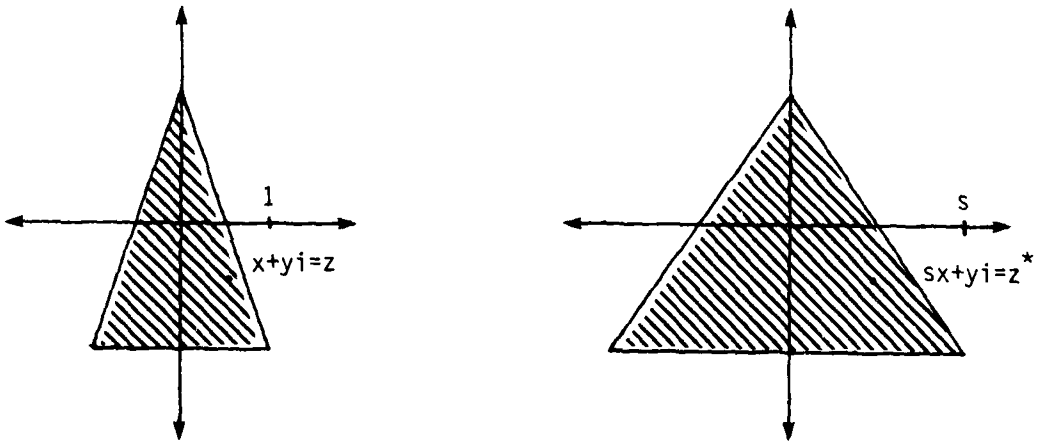


Figure 8. The Effect of Stretching a Picture Function by a Factor s in the X Direction.

By looking at our original definition of the centroid, $c^*(p)$, it should be clear that $I(1,0)/I(0,0)$ and $iI(0,1)/I(0,0)$ are the real and imaginary parts, respectively, of $c^*(p)$. Thus the above expression states that the centroid of the stretched picture function, $c^*(p^*)$, is obtained from the original centroid by multiplying its real part by the stretch factor s. Since we have assumed that $c^*(p) = 0$, it follows that $c^*(p^*) = 0$ as well. This simplifies the computation of higher moments:

$$\begin{aligned} M(2,c^*(p^*),p^*) &= M(2,0,p^*) \\ &= -sI(0,2) + 2is^2I(1,1) + s^3I(2,0) && \text{by (9)} \\ S(c^*(p^*),p^*) &= S(0,p^*) \\ &= s\int\int s^2x^2 + y^2dxdy \\ &= s^3I(2,0) + sI(0,2) \end{aligned}$$

so, $$S(c^*(p^*),p^*)^2 = s^6I(2,0)^2 + 2s^4I(2,0)I(0,2) + s^2I(0,2)^2$$

and $$|M(2,c^*(p^*),p^*)|^2 = s^3I(2,0)^2 - 2s^4I(2,0)I(0,2) + s^2I(0,2)^2 + 4s^4I(1,1)^2$$

Therefore, $S(c^*(p^*),p^*)^2 - |M(2,c^*(p^*),p^*)|^2 = 4s^4(I(2,0)I(0,2) - I(1,1)^2)$

Because the picture function p^* is identical with p when s is set equal to 1, it follows that:

$$S(c^*(p),p)^2 - |M(2,c^*(p),p)|^2 = 4(I(2,0)I(0,2) - I(1,1,)^2)$$

We use these relations with the formula for f^2 to obtain the desired result:

$$\begin{aligned} f^2(p^*) &= \frac{M(0,c^*(p^*),p^*)^4}{S(c^*(p^*),p^*)^2 - |M(2,c^*(p^*),p^*)|^2} \\ &- \frac{s^4M(0,c^*(p),p)^4}{s^4(S(c^*(p),p)^2 - |M(2,c^*(p),p)|^2)} \\ &= f^2(p) \end{aligned}$$

The fact that a picture may be stretched along any axis without modifying the value of f and f^2 explains some of our earlier findings. A rectangle and rhombus have the same value, 36, for f^2 since they can both be formed by stretching a square. It is easy to see that any parallelogram can be formed from a square in this manner so all parallelograms will have $f^2 = 36$. Similarly, any triangle can be formed from an equilateral triangle (a regular 3-gon) by two stretching operations and appropriate translation, rotation, and scaling. Since we have seen that f^2 for a binary equilateral triangle is 27, that value is the same for any triangle represented as a binary picture function. Applying the above result to our example of the aligned, equally spaced rectangles, we can see why the value of f^2 is independent of the rectangle's height: changing height is just a stretch operation.

The invariance of f^2 under stretching of the picture is important for another reason. Shapes in the plane appear as stretched when they are viewed from a position other than the perpendicular. A circle on the ground, for instance, is seen as an ellipse from an aircraft which is not directly overhead. Thus, for planar shapes, the invariance of f^2 under stretching is equivalent to the insensitivity of its shape recognition properties to the location of the viewer.

We close this section by reporting one last property of f and f^2. Suppose the picture function p is multiplied by the real number a to obtain a new picture function ap where $ap(x) = a(p(x))$, then $f(ap) = |a| f(p)$. The reader is invited to try his hand at demonstrating this relatively straightforward fact.

4.6 Generalized Balanced Ternary (GBT)

Until this point, we have dealt with picture functions that were defined at every point in the plane. Now we will deal with picture functions that are defined only on a discrete set of points in the plane. The discrete set we will be working with is the set of points that are addressable via the 2-dimensional Generalized Balanced Ternary (GBT) system. GBT is described at some length in the appendix to this report. The salient facts upon which the present development depends are stated below.

GBT is a spatial addressing system which supports the addressing of a fixed hierarchy of cells. The highest addressable cells are regular hexagons which we sometimes call zeroth level aggregates. The next higher cells all consist of seven hexagons grouped as one central hexagon and its six neighbors. These are called first level aggregates. In general, an nth level aggregate is composed of a central (n-1)st level aggregate and its six neighbors. Thus, an nth level aggregate contains 7^n hexagons. Figure 9 shows a third level aggregate and some of the addressing notation.

Since the distance between adjacent hexagon centers is 1, the inner radius must be ½. If follows that the outer radius of each hexagon must be $1/\sqrt{3}$ as shown in Figure 10.

A point is said to be addressable in GBT if it is at the center of any of the hexagons that make up the GBT grid. A correspondence between points addressable in GBT and a discrete subset of the complex numbers can be set up in the following way. The complex numbers 0 and 1 mapped to the centers of the hexagons whose GBT addresses are 0 and 1 respectively. The complex number i is mapped so that the vector from 0 to i is orthogonal to the vector from 0 to 1 in clockwise rotation. The point onto which i is mapped is not addressable in GBT. All other complex numbers are mapped linearly. The resulting correspondence between GBT addressable points within a first level aggregate and complex numbers is shown in Table 5.

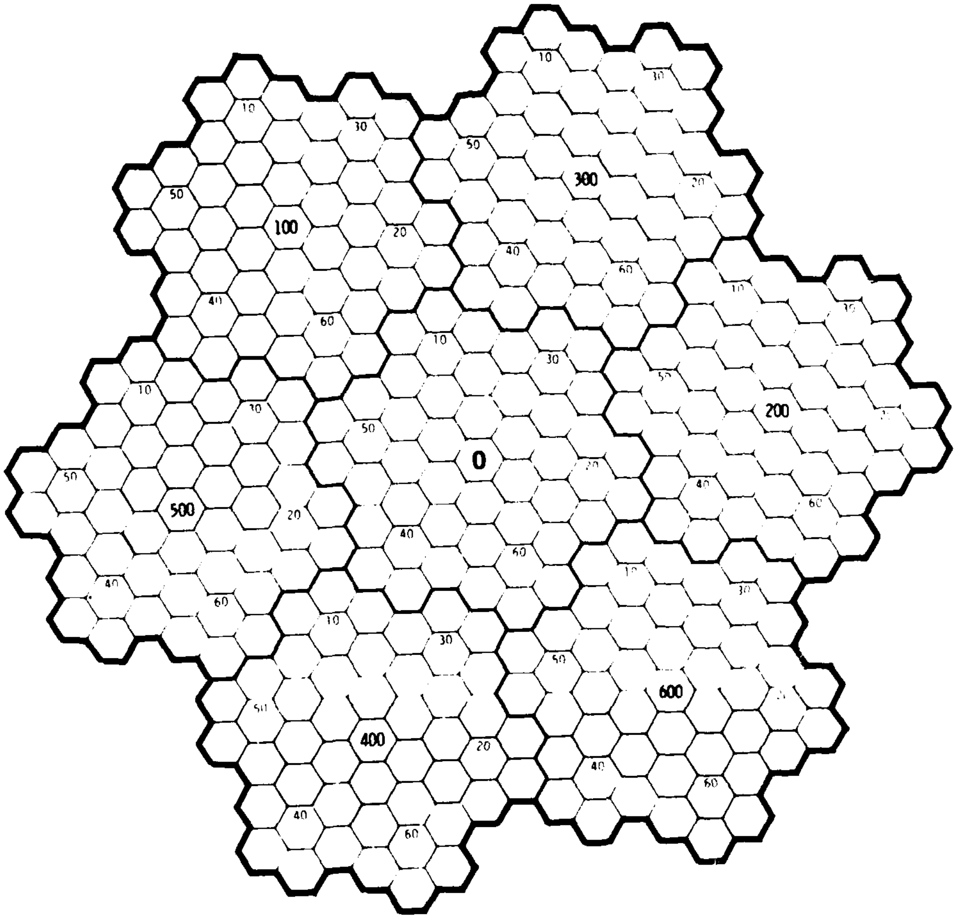


Figure 9. A Third Level GBT Aggregate.

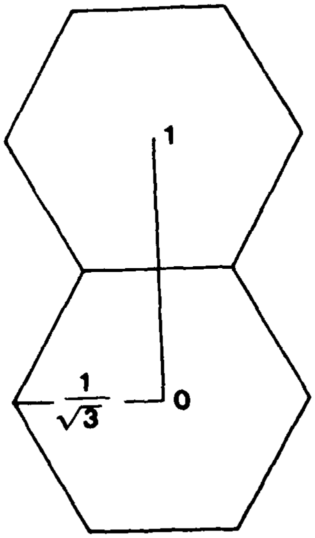


Figure 10. The Geometry of the GBT Hexagonal Grid.

Complex Numbers	GBT Addresses
0	0
1	1
$\frac{1}{2}(1 + \sqrt{3}i)$	3
$\frac{1}{2}(-1 + \sqrt{3}i)$	2
-1	6
$\frac{1}{2}(-1 - \sqrt{3}i)$	4
$\frac{1}{2}(1 - \sqrt{3}i)$	5

Table 5. The Correspondence between Addresses in the Central First Level Aggregate and Complex Numbers.

It turns out that the set of complex numbers which correspond to GBT addressable points under this mapping is closed under addition and multiplication. This means that GBT addressable points are a discrete subring of the complex numbers. The importance of this algebraic property derives from the fact that the various shape measures that we have described earlier depended upon the multiplication and addition (plus integration) of complex numbers. Thus, we have reasonable hopes for applying the same or similar formulas to GBT based discrete picture functions.

There are two other reasons for considering GBT based picture functions. One has to do with the technology for converting pictures to digital form. This is usually done by sampling the picture over a small region with some sort of scanning device and returning a single value that corresponds to an average of the picture function over that region. This value is generally viewed as being assigned to the centerpoint of the sample region. These values together with the points to which they are assigned are often called pixels (short for picture elements). Converting a picture function into pixels by a scanning process effectively discretizes the picture fuction. Thus, GBT, as a discrete system, provides a good framework for handling digitized images.

The last reason for considering picture functions based on GBT is the natural aggregation mechanism GBT provides. Our eventual goal is automated understanding of the content of images. A pixel, in itself, provides almost no information on the content of the image of which it is a part. Except for very simple cases, however, all of the pixels taken together provide too overwhelming a mass of complicated information to be easily comprehended by a computer. The GBT aggregate structure provides a sequence of intermediaries between these two extremes. The idea is that there must exist aggregates for a given location in a given picture that contain enough pixels so that significant entities can be discerned but not so many that the complexity is overwhelming. This theme of using aggregates to carve a picture up into understandable portions will be expanded upon considerably in the sections that follow.

4.7 G and H Functions

In this section we wish to define functions that are the discrete analogues of the complex moment and scalar moment functions that we used in the earlier sections. In what follows v is a variable GBT addressable point, c is a fixed complex number, p is a picture function defined on GBT addressable points, R is a region in the plane, and q and m are nonnegative integers. By analogy with the functions M and S of Section 4.2, we define:

$$G(q,c,p^m,R) = \sum_{v\in R} (v-c)^q p(v)^m$$

and

$$H(c,p,R) = \sum_{v\in R} |v-c|^2 p(v).$$

In order to use these functions to derive some of the useful shape descriptions we have been using in previous sections (e.g., eigenvalues and f^2), we must find their explicit relationship with M and S. The catch is that we do not know the values of p except at the GBT addressable points. We overcome this difficulty as follows. First, we define a new picture function $p^*(z)$ at each point $z\in C$ by $p^*(z) = p(v)$ where v is the closest GBT addressable point to z. This means that $p^*(z)$ has the same value as p(v) throughout the interior of the hexagon that contains v. That $p^*(z)$ is ill defined on hexagon boundaries does not matter in what follows since the hexagon boundaries are a set of measure zero. Second, we insist the region R be composed of whole hexagons. Thus if v is in or out of R then all points interior to the hexagon which contains v are in or out of R, *respectively.* Letting h(v) be the hexagon that contains v, it follows that:

$$\begin{aligned}
M(q,c,p^{*m},R) &= \int_{z\in R} (z-c)^q p^*(z)^m dz \\
&= \sum_{v\in R} \int_{z\in h(v)} (z-c)^q p^*(z)^m dz \\
&= \sum_{v\in R} p(v)^m \int_{z\in h(v)} ((z-v) + (v-c))^q dz \\
&= \sum_{v\in R} p(v)^m \sum_{k=0}^{q} \binom{q}{k} (v-c)^k \int_{z\in h(v)} (z-v)^{q-k} dz
\end{aligned}$$

Now the integral in this last expression is just the (q-k)th moment about its center of a regular 6-gon of outer radius $1/\sqrt{3}$. Its values are given as formula (3) of Section 4.3. Using this formula for q less than 6 we see that:

$$\int_{z\in h(v)} (z-v)^{q-k} dz = \begin{cases} \sqrt{3}/2 \text{ when } k = q \\ 0 \text{ otherwise} \end{cases}$$

Therefore, when q is less than 6, the qth moment simplifies as follows:

$$M(q,c,p^{*m},R) = \sum_{v\in R} p(v)^m (v-c)^q \sqrt{3}/2$$

$$= \sqrt{3}/2\ G(q,c,p^m,R).$$

As moments higher than 2 are not used in deriving our current shape descriptors, this result is all we need. The relationship of the S function to G and H functions is derived as follows:

$$S(c,p*,R) = \int_{z\epsilon R} |z-c|^2 p^*(z)dz$$

$$= \sum_{v\epsilon R} p(v) \int_{z\epsilon h(v)} |(z-v) + (v-c)|^2 dz$$

$$= \sum_{v\epsilon R} p(v) \int_{z\epsilon h(v)} |z-v|^2 + |v-c|^2 + (\bar{z}-\bar{v})(v-c)$$

$$+ (z-v)(\bar{\ }-\bar{c})dz$$

$$= \sum_{v\epsilon R} p(v)\ (\int |z-v|^2 dz + |v-c|^2 \int dz$$

$$+ (v-c)\int(\bar{z}-\bar{c})dz + (\bar{v}-\bar{c})\int(z-c)dz).$$

The latter two terms in this expression vanish since the first moment of a hexagon about its center (and its complex conjugate) are zero. The integral in the term with $|v-c|^2$ is $\sqrt{3}/2$ as we have seen above. The first integral can be evaluated by formula (4) of Section 4.3:

$$\int_{z\epsilon h(v)} |z-v|^2 dz = \frac{5}{24\sqrt{3}}$$

Therefore,

$$S(c,p^*,R) = \sum_{v\epsilon R} p(v)(\frac{5}{24\sqrt{3}} + |v-c|^2 \frac{\sqrt{3}}{2})$$

$$= \frac{\sqrt{3}}{2} H(c,p,R) + \frac{5}{24\sqrt{3}} G(0,c,p,R)$$

$$= \frac{\sqrt{3}}{2}(H(c,p,R) + \frac{5}{36} G(0,c,p,R))$$

We use the above expressions for M and S in terms of G and H to find the new formulas for our family of shape descriptors.

$$c^*(p^*,R) = c + \frac{G(1,c,p,R)}{G(0,c,p,R)} = \frac{G(1,0,p,R)}{G(0,0,p,R)}$$

$$\lambda_1(p^*,R) = ½(\frac{H(c^*,p,R)}{G(0,c^*,p,R)} + \frac{5}{36} + \frac{|G(2,c^*,p,R|}{G(0,c^*,p,R)})$$

$$\lambda_2(p^*,R) = ½\left(\frac{H(c^*,p,R)}{G(0,c^*,p,R)} + \frac{5}{36} - \frac{|G(2,c^*,p,R)|}{G(0,c^*,p,R)} \right)$$

$$f^2(p^*,R) = \frac{3}{16}\,\frac{G(0,c^*,p,R)^2}{\lambda_1(p,R)\lambda_2(p,R)}$$

$$= \frac{\frac{3}{4}\, G(0,c^*,p,R)^4}{(H(c^*,p,R) + \frac{5}{36}\, G(0,c^*,p,R))^2 - |G(2,c^*,p,R)|^2}$$

4.8 Agglomeration

Suppose we want to compute the values for G and H functions for a picture function, p, about a point, c, in a region, R. Suppose further that R is a disjoint union of t regions, R_j $(j = 1, \cdot\cdot\cdot, t)$, and we know the values of G and H functions for p about a point c_j in each region R_j. We would like to use the values of the functions in the subordinate regions, R_j, to compute their values on R. The method for accomplishing this is derived below:

$$G(q,c,p^m,R) = \sum_{j=1}^{t} \sum_{v \in R_j} (v - c_j + c_j - c)^q\, p(v)^m$$

$$= \sum_{j=1}^{t} \sum_{v \in R_j} \sum_{k=0}^{q} \binom{q}{k} (v-c_j)^k (c_j-c)^{q-k} p(v)^m$$

$$= \sum_{k=0}^{q} \binom{q}{k} \sum_{j=1}^{t} (c_j-c)^{q-k} G(k,c_j,p^m,R_j)$$

$$H(c,p,R) - \sum_{j=1}^{t} \sum_{v \in R_j} |v - c_j + c_j - c|^2\, p(v)$$

$$= \sum_{j=1}^{t} \sum_{v \in R_j} (|v-c_j|^2 + |c_j-c|^2 + (\bar{v}-\bar{c}_j)(c_j-c)$$

$$+ (v-c_j)(\bar{c}_j-\bar{c}))p(v)$$

$$= \sum_{j-1}^{t} (H(c_j,p,R_j) + |c_j-c|^2\, G(0,c_j,p,R_j)$$

$$+ (c_j-c)\, \bar{G}(1,c_j,p,R_j) + (\bar{c}_j-\bar{c})\, G(1,c_j,p,R_j))$$

The utility of the above formulas is that they provide a means of computing our family of shape descriptors for a region without having to deal with the potentially large volume of pixels within that region. All that is required is the knowledge of the values of appropriate G and H functions on a set of regions R_j that partition R. This feature is particularly well suited to computations within the GBT aggregate hierarchy.

Suppose our goal is to compute useful moments about the centers of GBT aggregates. Inductively, we can assume that we have computed all such moments for aggregates at level n-1 and below. To compute the moments at level n, we have only to use the formulas derived above based on the partition of each level n aggregate into its seven level n-1 subordinates. The computations are further simplified since the terms c_j-c are just the offsets of the centers of the subordinates from the center of the parent aggregate and are therefore known a priori. In the implementation work, this was the principle used for computing moments of higher cells. It is far more efficient than computing all moments based on pixels.

4.9 More Examples

To illustrate the principles discussed above, let us compute our family of shape descriptors for a picture function, p, that assigns the value 1 to all GBT addressable points within an nth level aggregate, A(n), and 0 to all other GBT addressable points. Accordingly let A_j, j = 0, $\cdots$, 6, be the 7 immediate subordinates of A(n), let c(n) be the center of A(n), and c_j the center of A_j. Then,

$$G(q,c(n),p,A(n)) = \sum_{k=0}^{q} \binom{q}{k} \sum_{j=0}^{6} (c_j-c(n))^{q-k} G(k,c_j,p,A_j)$$

$$= \sum_{k=0}^{q} \binom{q}{k} G(k,c(n-1),p,A(n-1)) \sum_{j=0}^{6} (c_j - c(n))^{q-k}.$$

The latter expression follows from the observation that all $G(k,c_j,p,A_j)$ are identical. But, for q-k less than 6:

$$\sum_{j=0}^{6} (c_j - c(n))^{q-k} = \begin{cases} 7 \text{ if } q-k = 0 \\ 0 \text{ otherwise} \end{cases}$$

This fact is established by an argument similar to that used for n-gons in Section 4.3. Thus, for q less than 6,

$$G(q,c(n),p,A(n)) = 7G(q,c(n-1),p,A(n-1))$$

$$= 7^n G(q,c(0),p,A(0))$$

But, $$G(q,c(0),p,A(0)) = \sum_{v \varepsilon A(0)} (v-c(0))^q p(v)$$

$$= (c(0) - c(0))^q$$

$$= \begin{cases} 1 \text{ if } q = 0 \\ 0 \text{ otherwise} \end{cases}$$

Therefore,

$$G(q,c(n),p,A(n)) = \begin{cases} 7^n \text{ if } q = 0 \\ 0 \text{ if } 0 < q < 6. \end{cases}$$

In particular we have,

$$M(0,c(n),p,A(n)) = \frac{\sqrt{3}}{2} 7^n$$

$$M(1,c(n),p,A(n)) = 0$$

$$c^*(p,A(n)) = c(n)$$

and $$M(2,c^*,p,A(n)) = M(2,c(n),p,A(n)) = 0.$$

We apply similar logic for the H function:

$$H(c(n),p,A(n)) = \sum_{j=0}^{6} (H(c_j,p,A_j) + |c_j-c(n)|^2 G(0,c_j,p,A_j)$$

$$+ (c_j-c)\, \bar{G}(1,c_j,p,A_j) + (\bar{c}_j-\bar{c})\, G(1,c_j,p,A_j))$$

The last two terms vanish since gradients of full aggregates are zero. Each $|c_j-c(n)|^2$ is the square of the distance from the center of a level n aggregate to the center of its jth level n-1 subordinate. For the central subordinate this distance is zero. For the six others, each $|c_j-c(n)|^2 = 7^{n-1}$. We have seen above that $G(0,c(n-1),p,A(n-1)) = 7^{n-1}$. Therefore,

$$H(c(n),p,A(n)) = 7H(c(n-1),p,A(n-1)) + 6 \times 7^{2n-2}$$

This is a classical recursion of the form $r(n) = ar(n-1) + bc^n$ whose solution is

$$r(n) = a^n r(0) + b\, \frac{a^n c - c^{n+1}}{a-c}.$$

Applying this solution with $a = 7$, $b = 6 \times 7^{-2}$, and $c = 7^2$, we find

$$H(c(n),p,A(n)) = 7^n H(c(0),p,A(0)) + 7^{n-1}(7^n-1)$$

$$= 7^{n-1}(7^n-1) \qquad \text{since } H(c(0),p,A(0)) = 0.$$

Applying the formulas for eigenvalues and f from Section 4.7 we find:

$$\lambda_1(p,A(n)) = \lambda_2(p,A(n)) = ½\, \left(\frac{H(c(n),p,A(n))}{G(0,c(n),p,A(n))} + \frac{5}{36}\right)$$

$$= ½\left(\frac{7^n-1}{7} + \frac{5}{36}\right) = \frac{36 \times 7^n - 1}{2 \times 7 \times 6^2}$$

$$f = \frac{wt}{2(\lambda_1 \lambda_2)^{\frac{1}{2}}} = \frac{18 \times \sqrt{3} \times 7^{n+1}}{36 \times 7^n - 1}$$

Values for f and f^2 for aggregates of levels 0 through 10 are given in Table 6. Notice that f^2 on a level zero aggregate is 38.88 as we would expect from our previous work on n-gons. Judging from the entries in the table, the values seem to be decreasing with n and converging. By checking the derivative of f and taking the limit, the reader can confirm that f decreases as n increases and converges to $7\sqrt{3}/2$ (f^2 to 36.75) as n goes to infinity.

n	f	f^2
0	6.23538	38.88000
1	6.08633	37.04341
2	6.06562	36.79170
3	6.06267	36.75595
4	6.06225	36.75085
5	6.06219	36.75012
6	6.06218	36.75002
7	6.06218	36.75000
8	6.06218	36.75000
9	6.06218	36.75000
10	6.06218	36.75000

Table 6. Values of f and f^2 for a picture function which is one within a level n GBT aggregate and zero elsewhere.

As a final example we have computed the value of f^2 for all 28 rotationally distinct first level GBT aggregates and indicated them in Figure 11. The two aggregates containing a single darkened hexagon have $f^2 = 38.88$ as we would expect from our results on n-gons in Table 3 and our results on level zero aggregates in Table 6. The completely filled first level aggregate has $f^2 = 37.04$ as expected from Table 6. For the special case of first level aggregates, f^2 differentiates completely between connected and disconnected patterns. Notice that each aggregate in Figure 11 is connected or disconnected if its f^2 value is greater or less than 15. An examination of the various shapes and f^2 values gives some insight as to the behavior of f^2 on more general binary patterns.

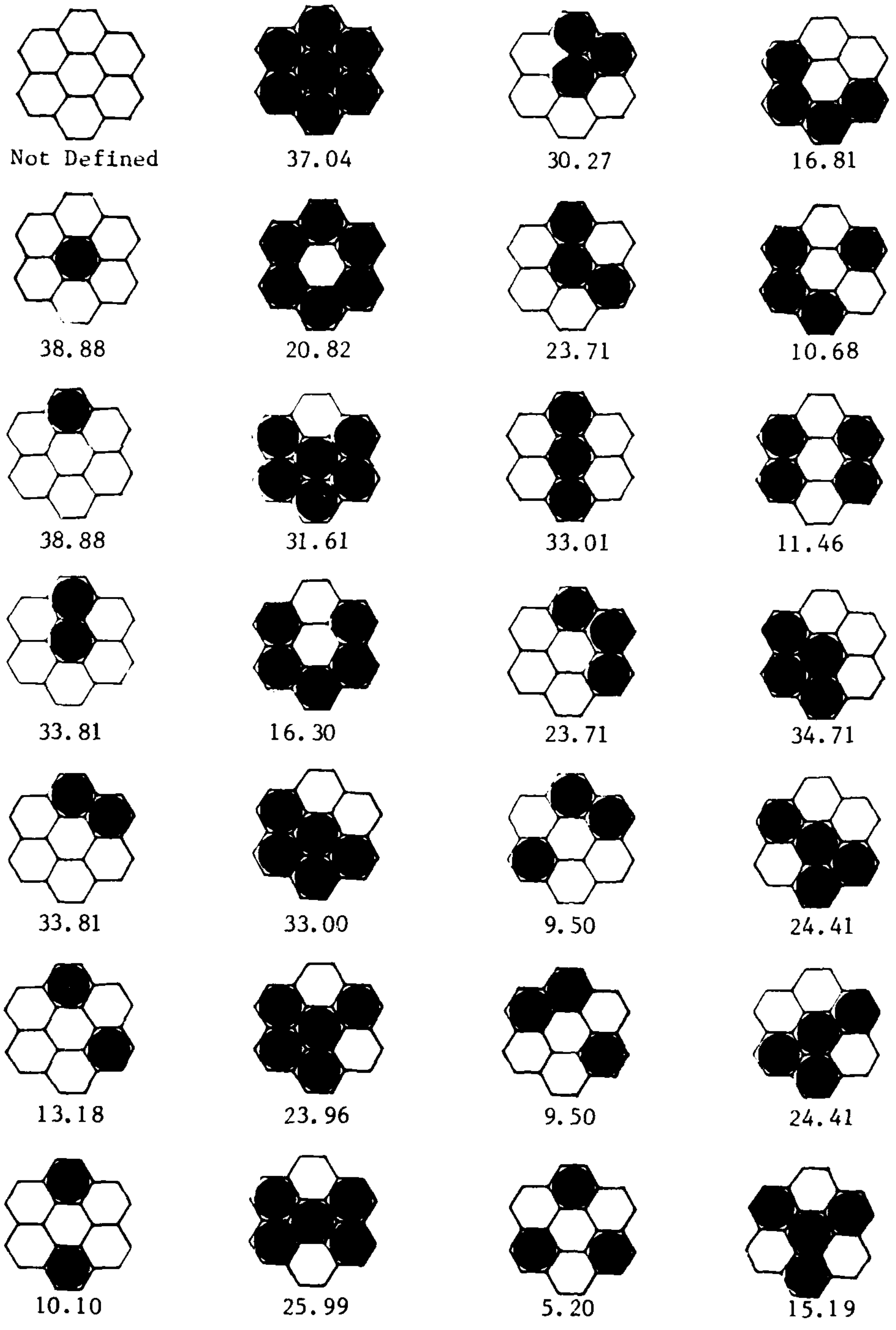


Figure 11. Values of f^2 for the 28 Rotationally Distinct Classes of Binary Picture Functions on First Level GBT Aggregates.

4.10 Texture

The shape descriptors that have been presented in the preceeding sections are quite powerful in identifying and separating objects from their background in an image provided the picture function has the value 1 on the objects and 0 on the background. In most applications, however, this is not the case. In a photograph, for instance, what our eye and brain recognize as a single entity may be made up of pixels of many different values. The human mind classifies different regions as being a part of the same entity, not because the pixels in the regions all have the same value, but because the regions have the same overall appearance. Let us refer to the notion of the general appearance of a region of a picture as its texture. In this section we wish to present a formal definition of the texture of a discrete, GBT based picture function in a region R. In the chapter on implementation we will show how this definition relates to our human notion of visual texture.

In what follows, let A(n) be an nth level GBT aggregate with immediate subordinates A_j, $j = 0, \cdot \cdot \cdot, 6$. We will use $A(k) < A(n)$ to say that A(k) is a kth level aggregate that is contained in A(n). First let us relate mean and variance to G functions. Recalling the definitions from Section 4.2 and relations of Section 4.7,

$$\mu(p,A(n)) = \frac{M(0,c,p,A(n))}{M(0,c,p^0,A(n))} = \frac{G(0,c,p,A(n))}{G(0,c,p^0,A(n))}$$

$$= \frac{1}{7^n} G(0,0,p,A(n))$$

$$Var(p,A(n)) = M(0,c,p^2,A(n))/M(0,c,p^0,A(n)) - \mu(p,A(n))^2$$

$$= \frac{1}{7^n} G(0,0,p^2,A(n)) - \frac{1}{7^{2n}} G(0,0,p,A(n))^2$$

Another useful concept, the local variance, Var^*, of p on A(n) is defined by:

$$Var^*(p,A(n)) = \frac{1}{7} \sum_{j=0}^{6} \mu(p,A_j)^2 - \mu(p,A(n))^2.$$

The local variance ignores the fact that there is variability within the immediate subordinates A_j of A(n). It would be the true variance if p was uniform on each A_j. Suppose v is a GBT addressable point and we have a picture function p defined within a universe that consists of a level m GBT aggregate that contains v. Then we define the texture of p at v as follows:

$$T(p,v) = (Var^*(p,A(1)),\ Var^*(p,A(2)),\ \cdot \cdot \cdot,\ Var^*(p,A(m)))$$

where each A(k) is the unique kth level aggregate containing v. For a region R we define the texture of p in R by:

$$T(p,R) = \frac{1}{n(R)} \sum_{v \in R} T(p,v)$$

where n(R) is the number of GBT addressable points v in R. Suppose R is a disjoint union of L regions $R_1, \cdot \cdot \cdot, R_L$. Then,

$$T(p,R) = \frac{1}{n(R)} \sum_{j=0}^{L} \sum_{v \in R_j} T(p,v)$$

$$= \sum_{j=0}^{L} \frac{n(R_j)}{n(R)} T(p,R_j)$$

This says that the texture of a region partitioned into subregions can be written as a convex sum of the subregion textures each weighted by the relative size of the subregion.

Applying this result to an nth level aggregate, A(n), we get:

$$T(p,A(n)) = \frac{1}{7} \sum_{j=0}^{6} T(p,A_j) \tag{10}$$

If we sum up the first n components of T(p,A(n)) the result is Var(p,A(n)). To demonstrate this result let us assume that it is true for all aggregates of level n-1 and below and proceed by induction:

$$\sum_{k=1}^{n} T(p,A(n))_k = \sum_{k=1}^{n} \frac{1}{7} \sum_{j=0}^{6} T(p,A_j)_k$$

$$= \frac{1}{7} \sum_{j=0}^{6} (\sum_{k=1}^{n-1} T(p,A_j)_k + Var^*(p,A(n))$$

$$= \frac{1}{7} \sum_{j=0}^{6} Var(p,A_j) + Var^*(p,A(n))$$

$$= \frac{1}{7^n} \sum_{j=0}^{6} G(0,0,p^2,A_j) - \frac{1}{7^{2n-1}} \sum_{j=0}^{6} G(0,0,p,A_j)^2$$

$$+ \frac{1}{7} \sum_{j=0}^{6} \frac{1}{7^{2n-2}} G(0,0,p,A_j)^2 - \frac{1}{7^{2n}} G(0,0,p,A(n))^2$$

$$= Var(p,A(n))$$

The definition of texture provided here makes the texture at a GBT addressable point a function of the behavior of the picture function in all of its superior aggregates. Since this behavior cannot be determined by looking at the point alone, there is some question as to how one computes texture. The method is suggested by formula (10). Suppose, inductively, that we have computed the first k components of the texture vector for each kth level aggregate for k $\leq$ n-1 and stored it at that aggregate address together with the mean for that address. Then we can compute the first n components of the texture for A(n) as follows:

$$\text{for } k < n, \quad T(p,A(n))_k = \frac{1}{7} \sum_{j=0}^{6} T(p,A_j)_k$$

$$T(p,A(n))_n = \text{Var}^*(p,A(n))$$

$$= \frac{1}{7} \sum_{j=0}^{6} \mu(p,A_j)^2 - \frac{1}{49} \left(\sum_{j=0}^{6} \mu(p,A_j) \right)^2$$

Proceeding by this method, each nth level aggregate knows the first n components of its texture vector. Since the GBT data access scheme requires that each superior aggregate of an nth level aggregate, A(n), be traversed to reach the data associated with A(n), the texture components higher than n can be retrieved during this operation. In this manner the texture of any aggregate at any level can be found in a manner natural to the GBT data management process.

The discussion of texture provided above leaves one major question unanswered. What does this mathematical formulation have to do with our human notion of visual texture? A preliminary answer is provided in the next chapter.

5.0 IMPLEMENTATION

In the previous chapter the notion of visual texture was given formal mathematical definition. In addition, a family of measures which describe the behavior of picture functions over areas was developed. In the case of texture, it was necessary to compute the texture for a variety of samples to see how this formal definition related to intuitive ideas about texture. Aggregate descriptors were studied by using them in one image analysis application, raster to vector conversion. The work that was done attempted to find and demonstrate the utility of the theoretical constructs of Chapter 4.0 to problems of understanding imagery. The results, while preliminary, indicate the richness of this approach. They are described in this chapter.

5.1 Texture

In order to examine texture, a set of sample images was collected. These were photocopied from photographs to insure uniformity of resolution in the image media. The samples were scanned at a 100 micron resolution using an Optronics C-4100 Colorscan System. For each pixel v, a gray scale value, p(v), was assigned where p(v) is an integer and $0 \leq p(v) \leq 255$. Figures 12 through 16 are from electrostatic plots of these scans. The sample textures are labeled texture 1 through texture 5.

Each scan was of such an area that there were thirty-two level four aggregates completely filled with pattern. Those aggregates on the boundary which were not completely filled were ignored.

For a given texture, each full level four GBT aggregate was assumed to be a sample of the texture. Its texture vector was computed.

Suppose A is such a level four aggregate. In Section 4.10 the texture of A is defined to be $(V_1(p,A), V_2(p,A), V_3(p,A), V_4(p,A))$ where $V_k(p,A)$ is the mean of the local variances of the level k aggregates subordinate to A and the local variance of a level k aggregate is just the variance of the mean of p over each of its seven level k-1 subordinates. Table 7 gives the mean texture vector for each sample together with the standard deviation in each component. From these it can be seen that the vectors of each texture cluster together. To get some indication of the separation of these clusters, the Mahalanobis distance was used. This statistic measures the distance in terms of statistical variation to the mean vector of a multivariate normal distribution. It is defined:

$$d(v,\mu) = (v - \mu)^t \, \Sigma^{-1} \, (v - \mu)$$

where μ is the mean vector, Σ is the covariance matrix, and v is a vector. Table 8 shows the average, maximum, and minimum distances from the vectors of one texture to the mean of another. It was found that for samples 1, 2, and 4, if v is a vector of texture T and w is a vector of a different texture, v is closer to the mean of the T distribution than w is. While such a result does not hold in general (textures 3 and 5 do not have this property with respect to each other), it does indicate that samples of like texture form well-defined clusters.

	Mean	Standard Deviation
Texture 1	(234, 313, 191, 78)	(65, 86, 92, 40)
Texture 2	(270, 457, 695, 551)	(83, 95, 161, 272)
Texture 3	(623, 553, 533, 141)	(343, 82, 68, 56)
Texture 4	(476, 673, 200, 24)	(169, 64, 34, 14)
Texture 5	(1258, 746, 356, 48)	(418, 97, 76, 24)

Table 7. Mean Vectors and Standard Deviations for Sample Textures.

	Texture 1	Texture 2	Texture 3	Texture 4	Texture 5
Texture 1					
Average	1.8	13.8	14.2	7.9	33.5
Maximum	3.0	28.1	49.0	23.6	51.9
Minimum	0.7	8.0	3.5	3.8	3.3
Texture 2					
Average	8.8	1.9	6.5	8.2	13.5
Maximum	10.8	2.9	17.0	11.4	18.6
Minimum	6.3	1.1	3.6	7.3	6.0
Texture 3					
Average	13.8	12.4	1.8	7.6	4.7
Maximum	19.0	22.1	3.1	10.3	6.5
Minimum	6.7	7.1	0.8	6.0	2.7
Texture 4					
Average	8.5	51.1	17.9	1.9	8.5
Maximum	15.4	96.7	26.5	3.3	13.6
Minimum	3.7	31.6	12.0	1.2	6.2
Texture 5					
Average	6.2	25.3	6.0	4.3	1.9
Maximum	7.7	53.6	11.9	5.3	3.1
Minimum	3.2	11.6	2.5	3.1	1.2

Table 8. Mahalanobis Distance. A row contains the average, maximum, and minimum Mahalanobis distance to the mean of a texture as computed for all samples of the texture represented by the column.

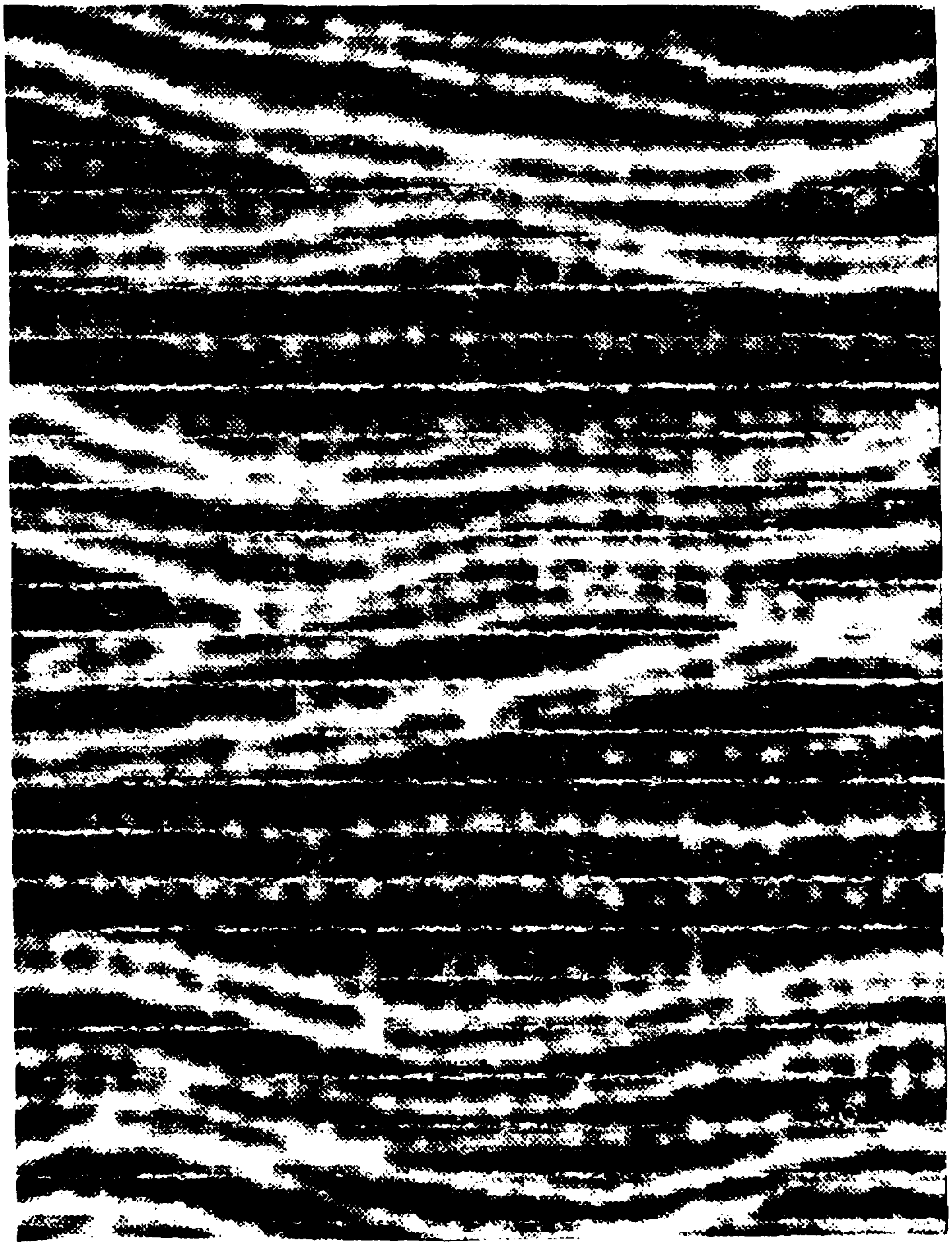


Figure 12. Sample Texture 1. From a Photograph of Loose Woven Fabric.

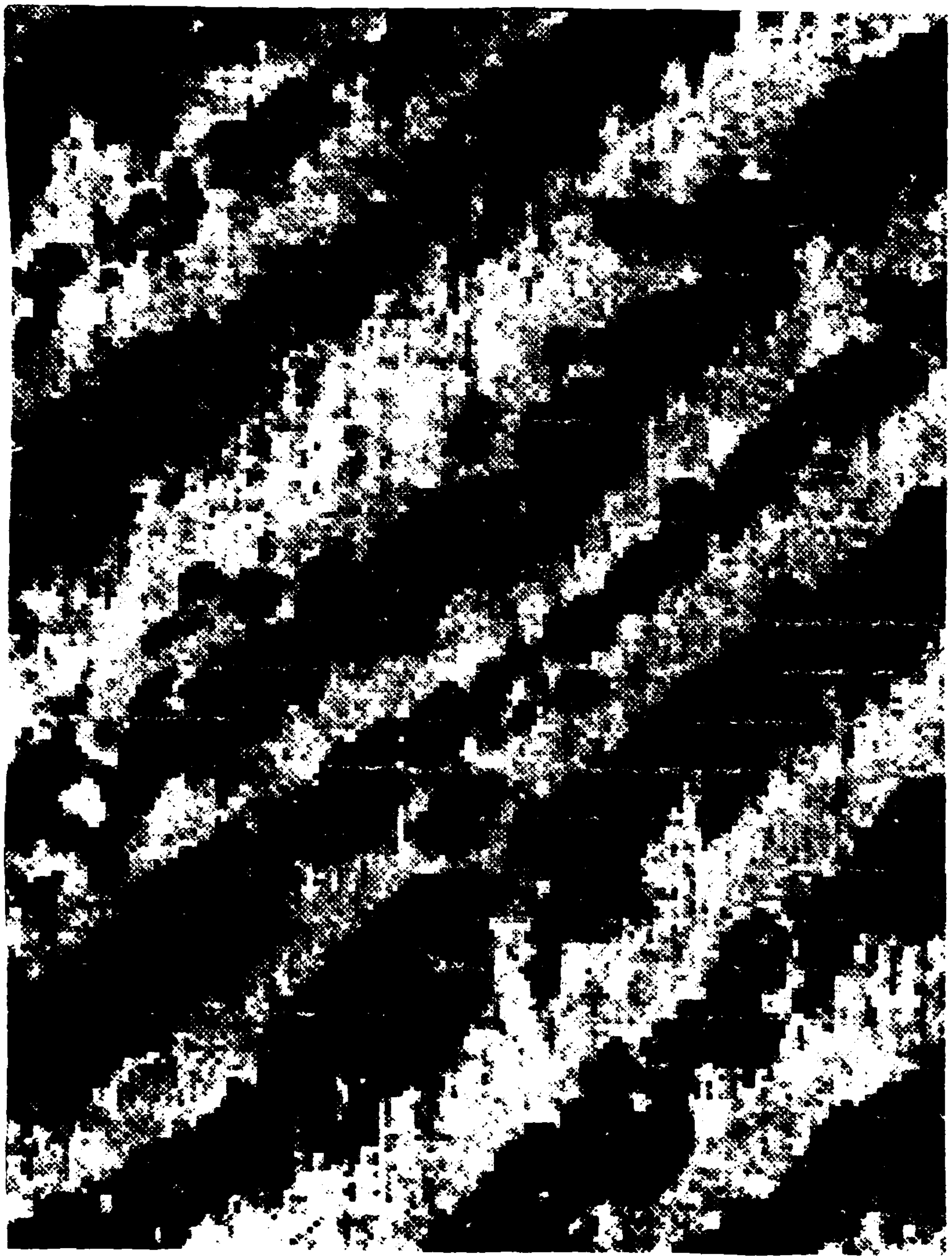


Figure 13. Sample Texture 2. From a Photograph of Pleated Fabric.

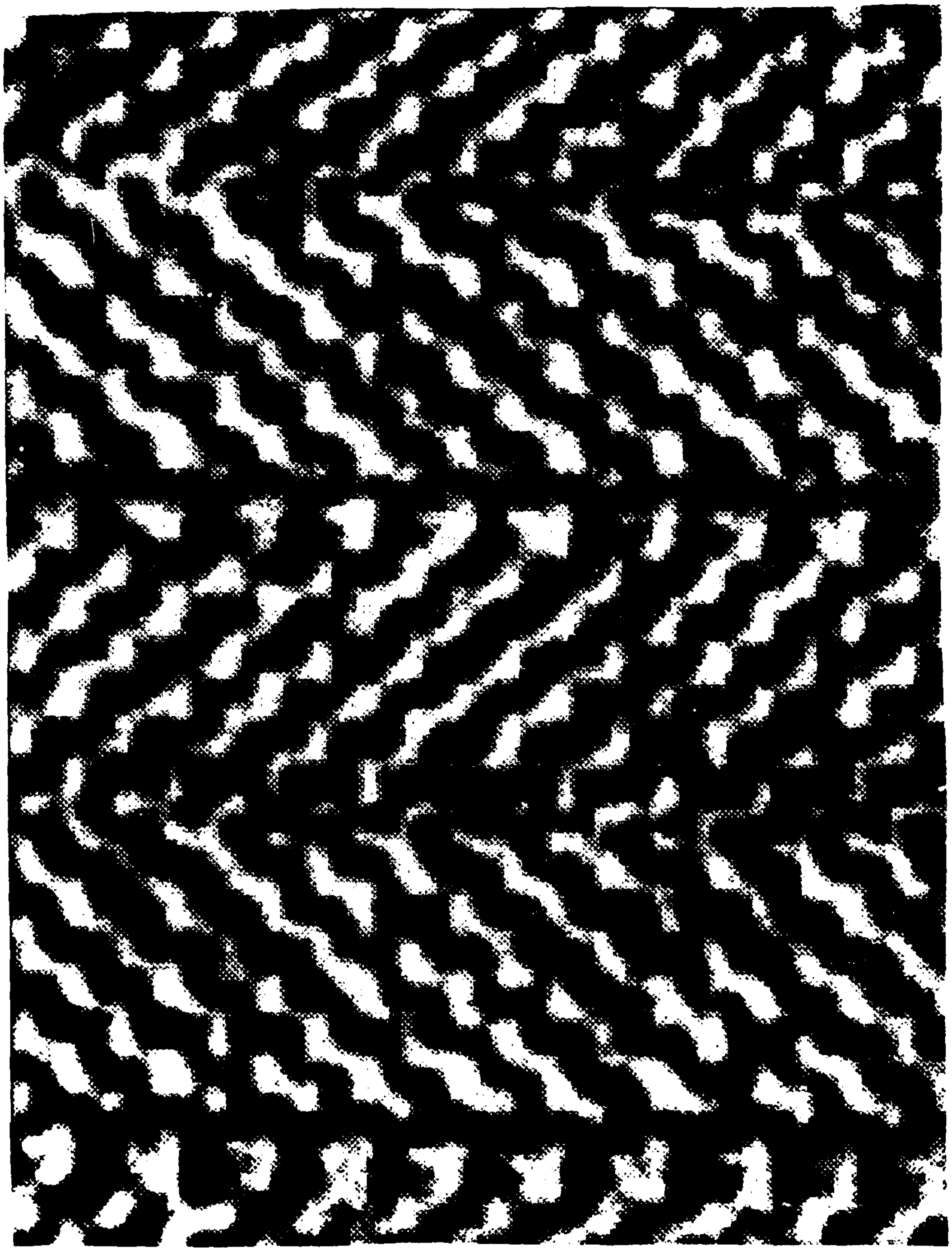


Figure 14. Sample Texture 3. From a Photograph of Herringbone Wool Fabric.

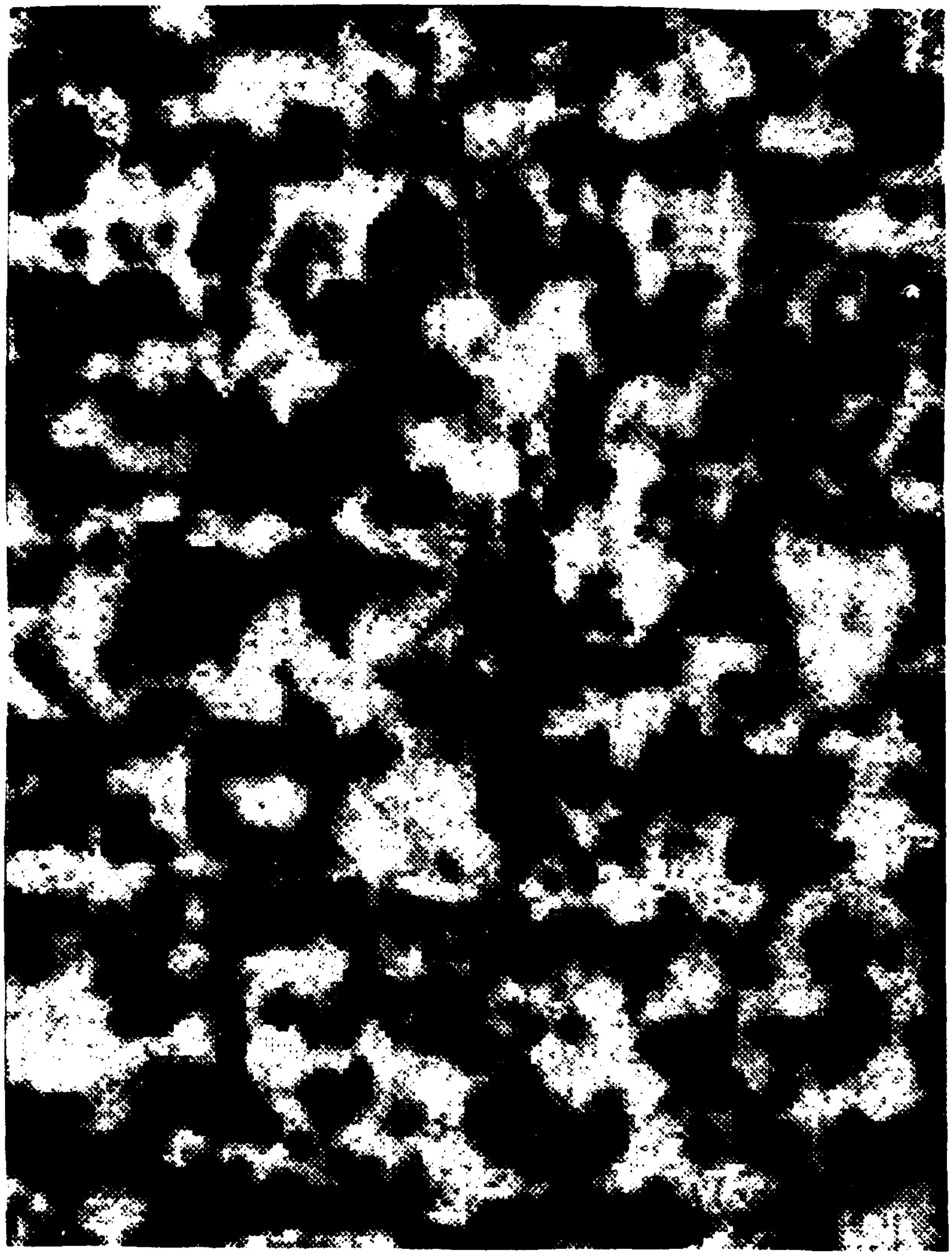


Figure 15. Sample Texture 4. From a Photograph of Nubby Wool Fabric

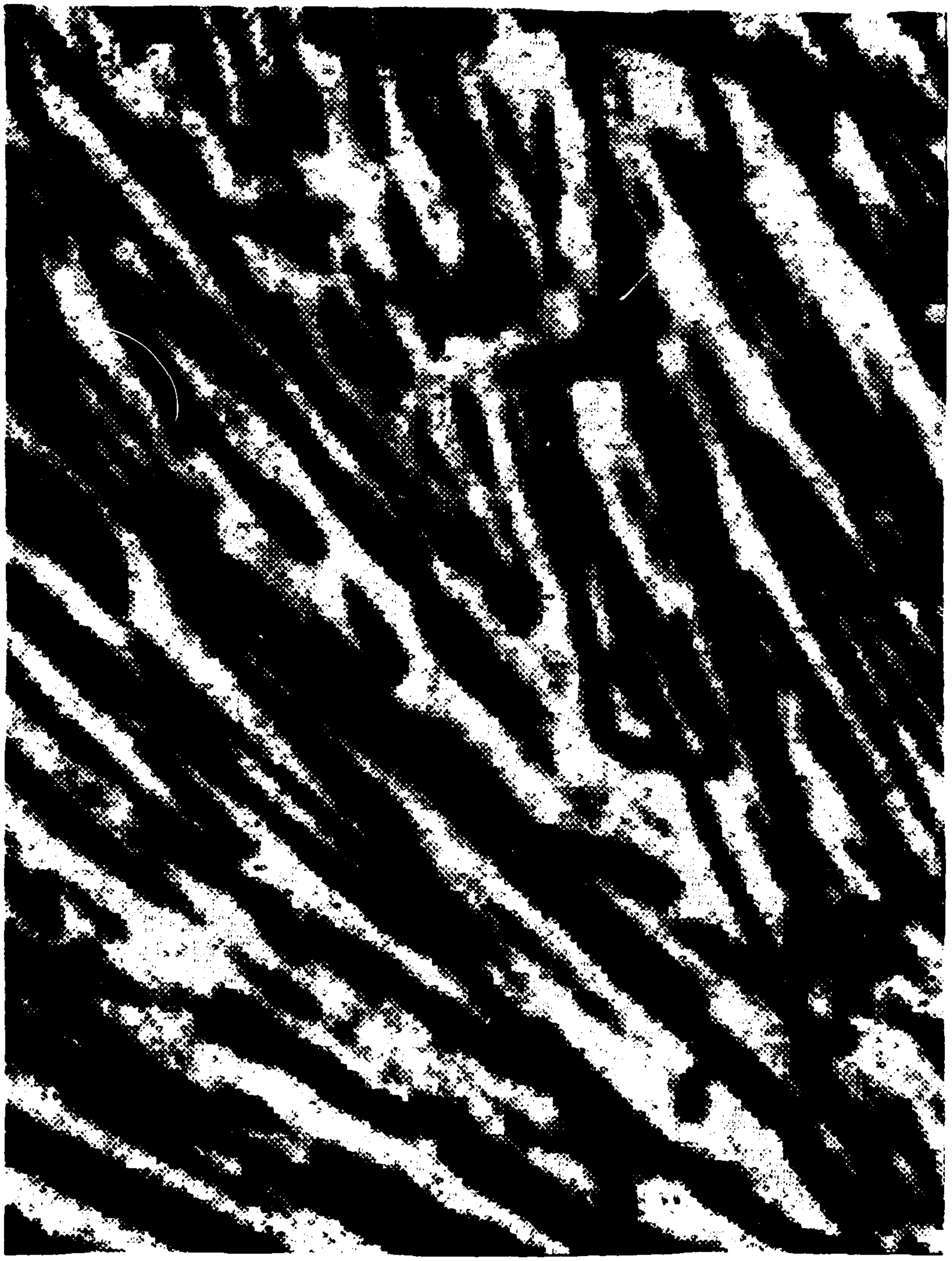


Figure 16. Sample Texture 5. From a Photograph of Straw.

5.2 Shape Descriptors

The work done with the family of aggregate descriptors defined in Section 4.0 used existing software for processing binary (0 or 1 gray scale valued) images. This software takes the output of a scan file and populates a GBT file. I was enhanced to compute and store the aggregate descriptors for all aggregates at levels one through four. This file can be accessed through a graphics device. Figures 17 through 21 are electrostatic plots of displays. The operator can also ask for the descriptors of a fourth level aggregate which is shown on the screen.

The descriptors were used in an algorithm designed to replace an original raster image with a vector representation. This algorithm can be outlined as follows:

1. Take a raster image and populate a GBT file.

2. Compute descriptors among them: weight, gradient, the eigenvalues λ_1 and λ_2, the angle of the second moment.

3. Classify each aggregate based on its descriptors. The classes are: line segment, line edge, full, and other.

4. Start with a level four aggregate A. If A is classed as a line segment, fit a line to A. If A is classed as a line edge find other edges, agglomerate the statistics, and fit a line to the rasters. If A is full, do nothing. If A is "other", look at its level three subordinates. Handle them according to their classifications. The result of this algorithm is a file of line pieces that can be displayed graphically to see how well the aggregate descriptors represent the raster patterns.

The examples described here are taken from a scan at a 50 micron resolution of an electrical circuit diagram. Figure 17 shows a horizontal line in that diagram in raster form with the GBT aggregates overlaid. Table 9 gives the values of some of the descriptors for the fourth level aggregate in the center of Figure 17. Table 10 shows the corresponding values for aggregate B in Figure 19 and for each of its seven subordinates.

Weight:	338.0
f^2:	34.9
λ_1:	203.7
λ_2:	3.0
Angle of Second Moment:	180°

Table 9. Descriptors for Level Four Aggregate A in Figure 17.

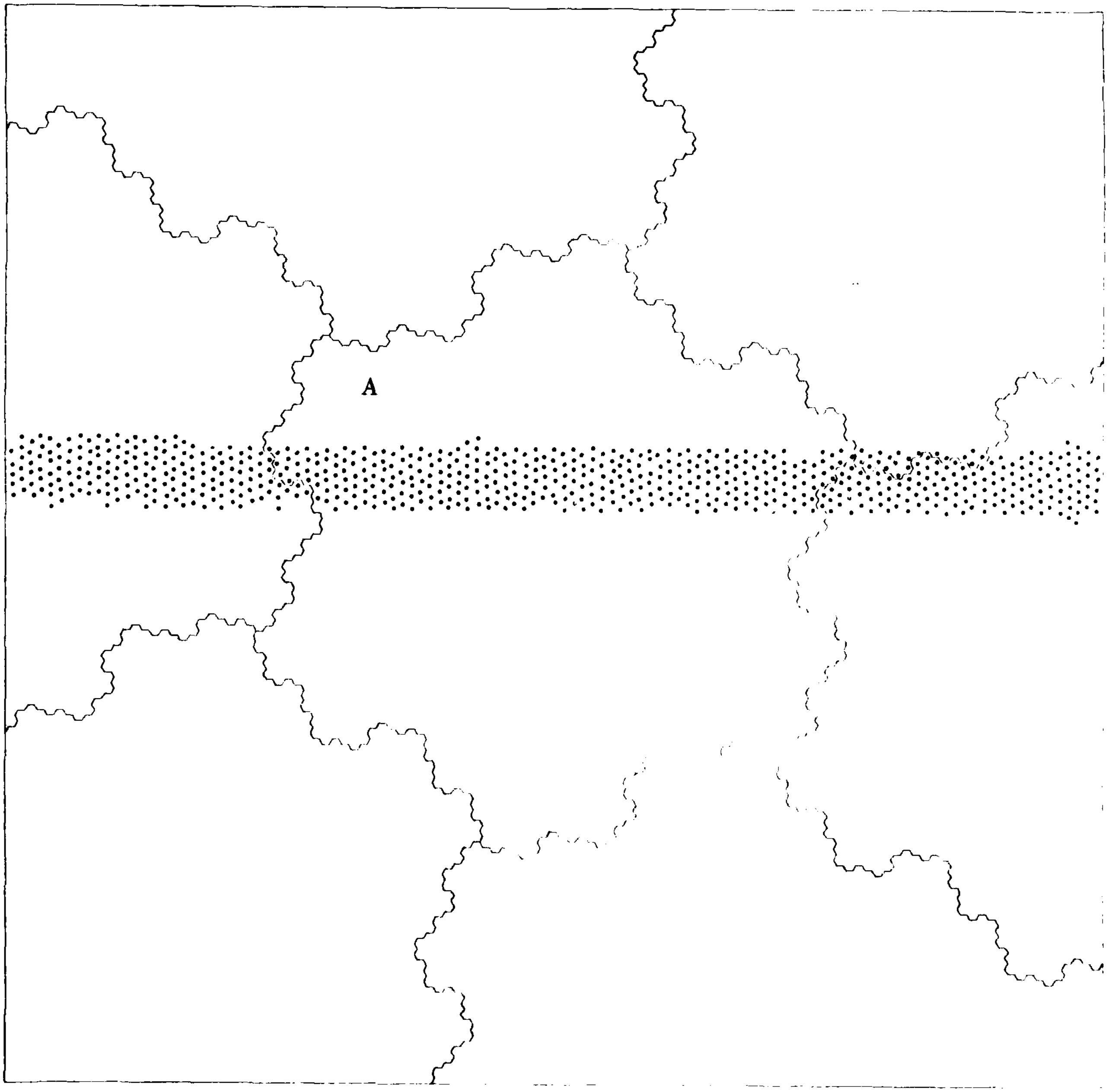


Figure 17. A Portion of a Raster Scanned Diagram. The Descriptors for the Fourth Level Aggregate A are given in Table 9.

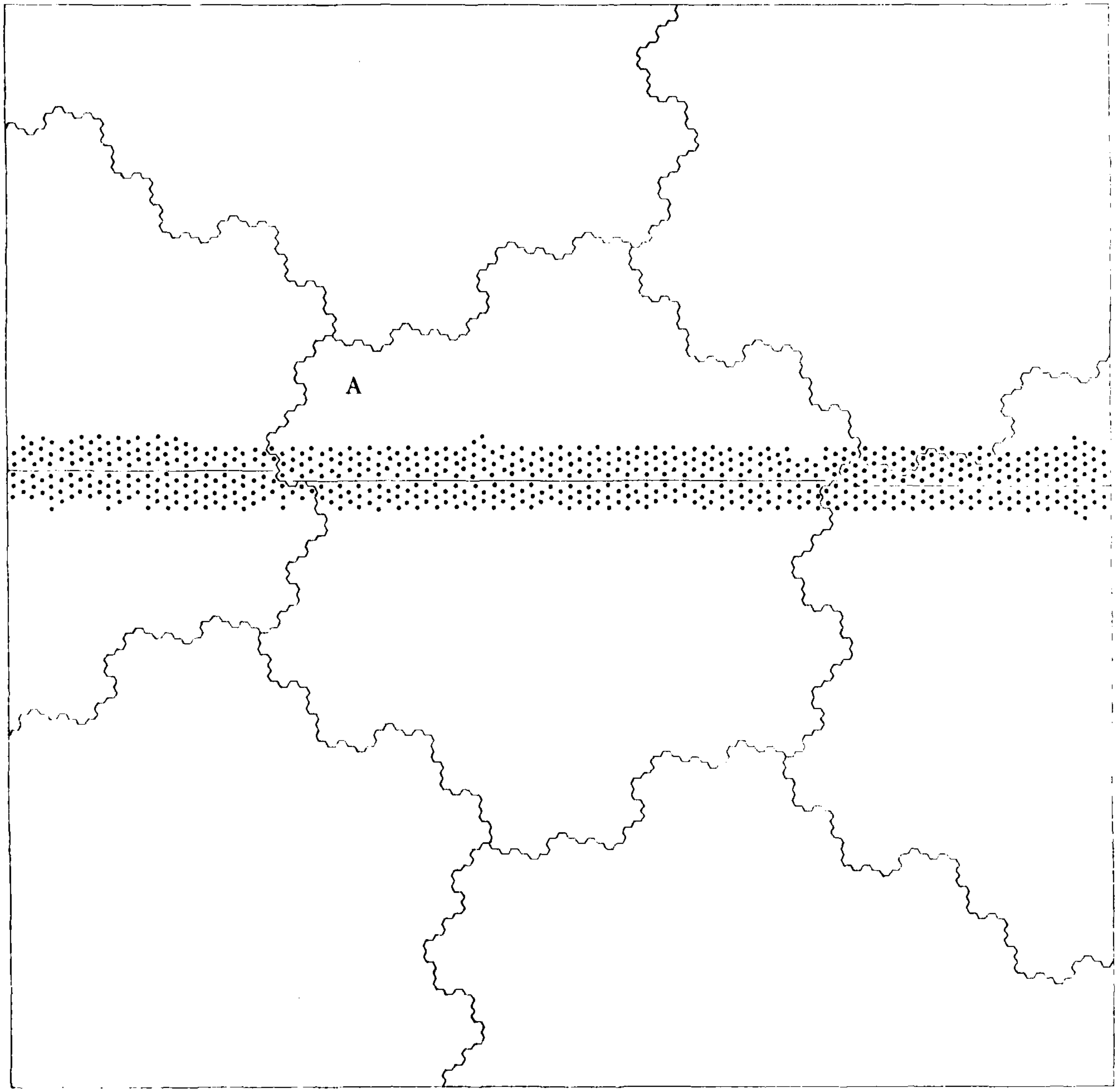


Figure 18. Based on Descriptors, a Line Segment is computed for the Aggregate A.

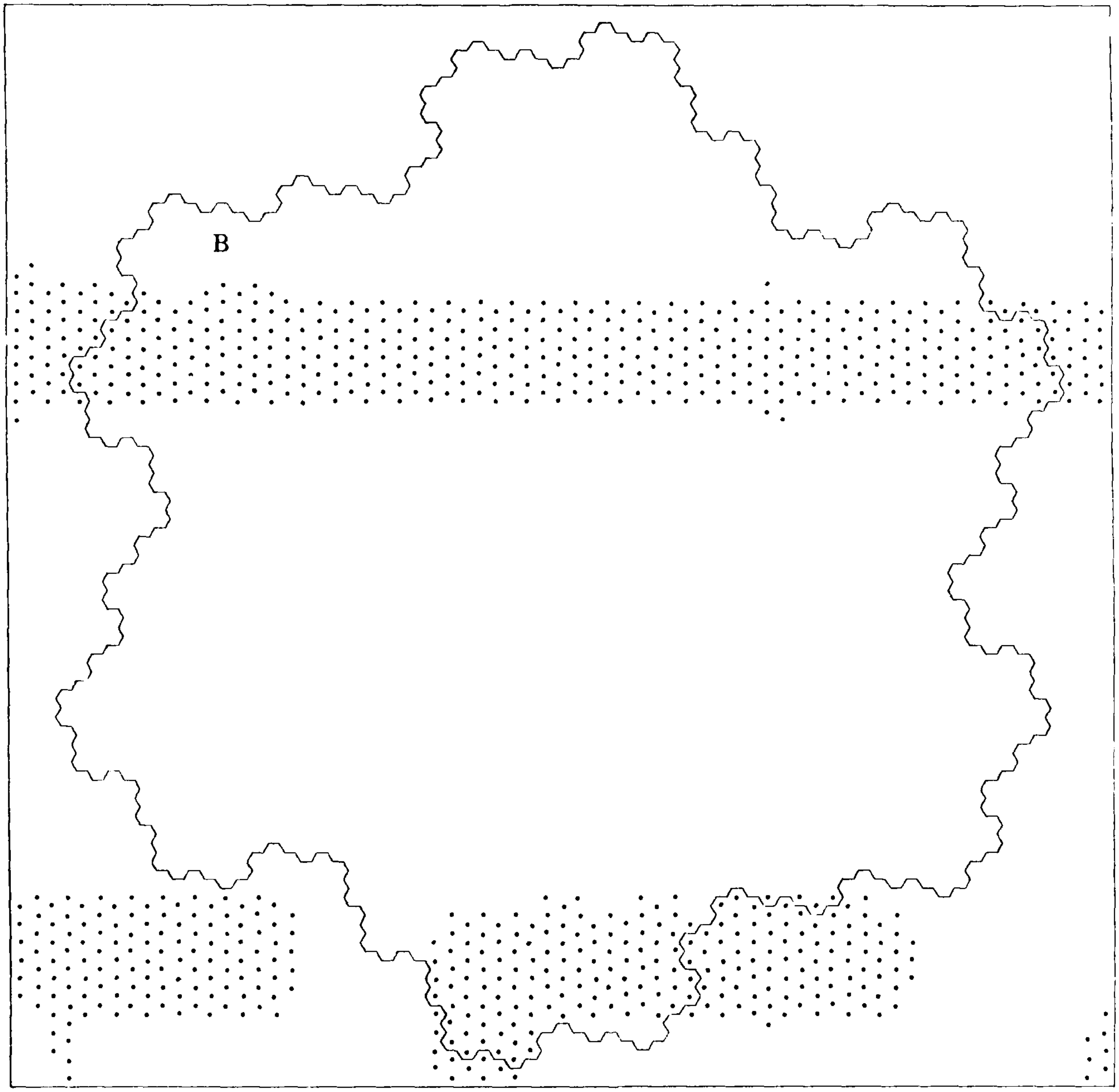


Figure 19. A Portion of a Raster Scanned Diagram. The Descriptors for the Fourth Level Aggregate B are given in Table 10.

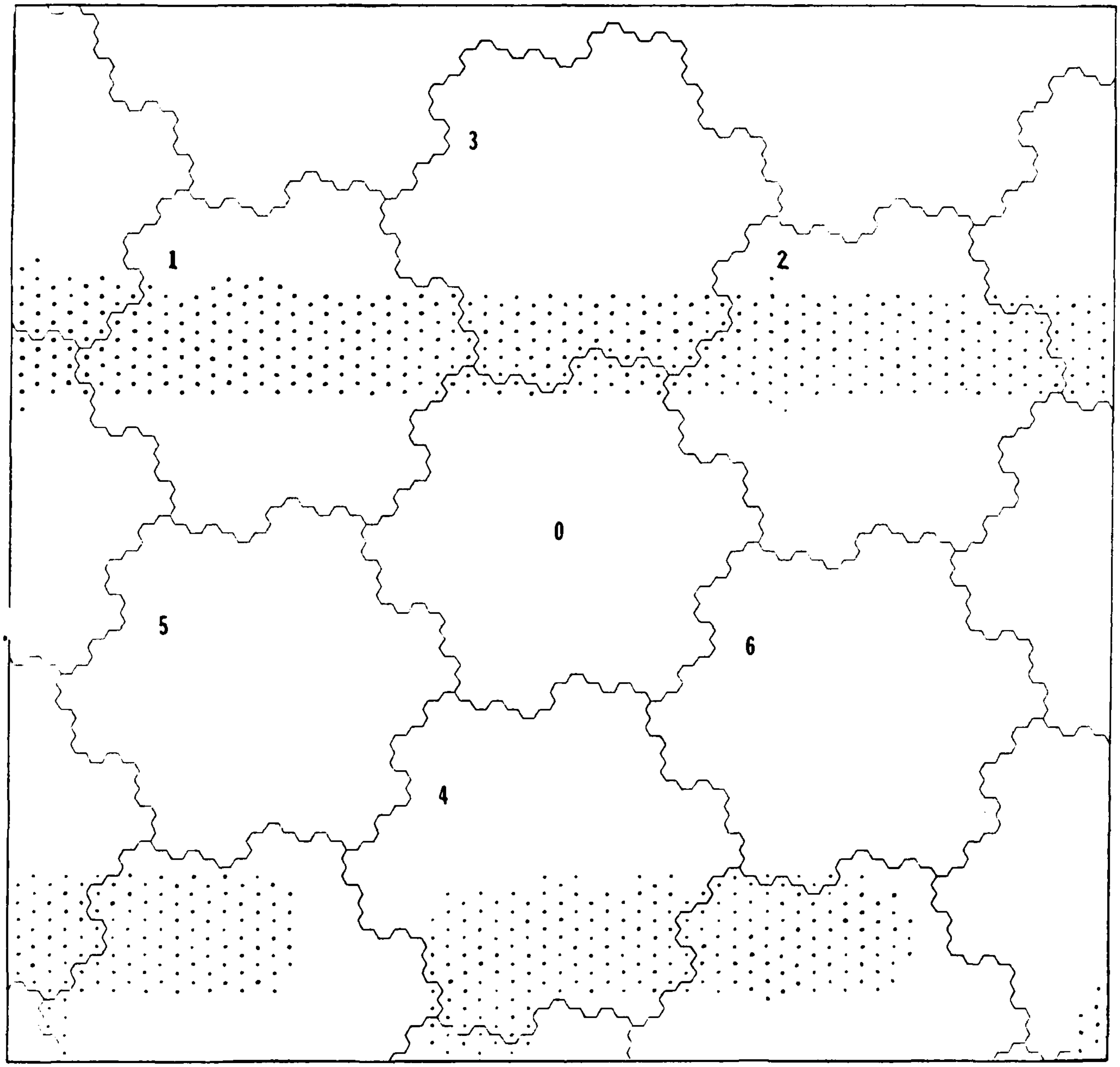


Figure 20. The Aggregate B and its Third Level Subordinates.

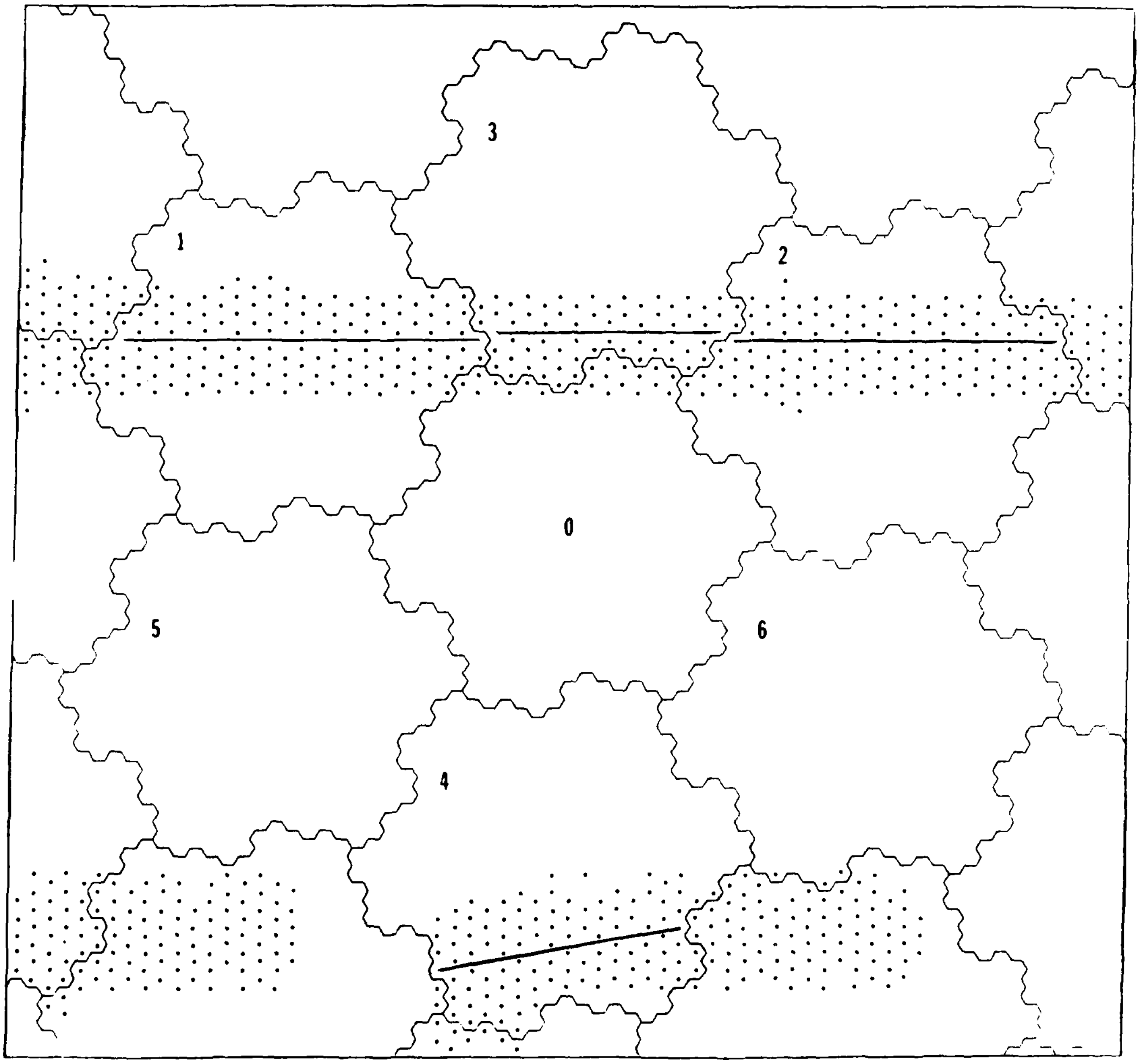


Figure 21. Based on Descriptors, Line Segments are computed for the Subordinates of B.

Aggregate B

Weight:	484.0
f^2:	1.2
λ_1:	22.16
λ_2:	171.4
Angle of Second Moment:	.78°

Subordinates	0	1	2	3	4	5	6
Weight:	21.0	142.0	130.0	70.0	117.0	0.0	4.0
f^2:	16.0	36.0	33.6	30.9	32.5	0.0	15.6
λ_1:	12.8	32.2	30.2	16.0	17.2	0.0	1.7
λ_2:	.4	3.3	3.1	1.9	4.6	0.0	.1
Angle of Second Moment:	175°	178°	179°	173°	158°	0°	185°

Table 10. Descriptors for Level Four Aggregate B in Figure 19 and for its Subordinates shown in Figure 20.

The line fitting algorithm processes aggregate A in this way:

1. Check Weight. A weight of 338 corresponds to 14% filled, neither a "full" aggregate nor so little as to make the other measures misleading.

2. Check f^2. An f^2 of 34.9 indicates a pattern which is connected and very rectangle-like (recall that f^2 for a rectangle is 36).

3. Consider λ_1 and λ_2. In this case they indicte a long, thin rectangle, a line segment.

4. The angle of the second moment is 180° (from the vertical). Thus the angle of the major axis is half 180° or 90°.

5. Approximate this pattern with a horizontal line segment passing through the centroid of the pattern. Figure 7 shows the line segment that was computed.

6. Since the scan resolution is 50 microns and $\lambda_2 = 3$, compute and store an average line width of .0012 inch.

This technique allows the rasters in this portion of the diagram to be replaced rather quickly with a concise description of the pattern which they represent. In this way, all of the simple areas of the diagram can be converted from raster form. The next example demonstrates how more complex regions are handled The algorithm processes aggregate B in this way.

1. Check Weight. A weight of 484 indicates 20% of the aggregate is filled, a significant portion.

2. Check f^2. f^2 is 1.2 for aggregate B. The pattern in B is divided into disjoint pieces. Because of this, B cannot be accurately represented with a single line. Classify B as "other" and process its seven subordinates.

3. Check Weight. Those subordinates with weight sufficient to be considered are 0, 1, 2, 3, and 4. Subordinates 5 and 6 are less than 5% filled.

4. Check f^2. Subordinates 1, 2, and 4 each have a high f^2: 36, 33.6, and 32.5. Subordinate 3 is not as high but still indicates a solid, connected piece.

5. Consider λ_1 and λ_2. In subordinates 1, 2, and 4 they indicate a long, thin shape from .0012 to .0018 inches wide. These subordinates are classified as line segments.

6. In the case of subordinate 3, because of its f^2 and λ_2 values, the centroid of the pattern is examined. It lies very close to the aggregate boundary. This subordinate is classified as a line edge.

7. The last step in this process tries to match subordinate 3 with "the other edge" of the line. In this case, 0 and 3 are agglomerated. The resulting statistics indicate a line segment which is computed.

Figure 21 shows the final results of this algorithm on aggregate B.

Preliminary work with the descriptors defined in Chapter 4.0, has shown that these descriptors can be used to understand complex images. The representations obtained merely by substituting line segments in aggregates have been much more accurate than previous vectorizations. This approach is also fast since large, simple areas of the image are processed at one time. More detailed computations are reserved for those areas where complexity is indicated.

6.0 FUTURE RESEARCH DIRECTIONS

6.1 Texture

The results on texture presented in this report provide a solid beginning for future research and development. Texture has been quantified; important properties of that quantification have been demonstrated; and the quantification has been shown to correspond to the intuitive idea of visual texture. What remains is to use this quantification of texture in computers to decompose complex images into relatively simple pieces so that shapes, properties, and relations of entities in the image can be more easily analyzed. This will require a major effort but the potential rewards are great.

Table II shows the status of various questions regarding needed follow on work on texture.

Question	Answer	Comment
Can texture be defined for a discrete picture function in a region of GBT addressable points?	Yes. And the texture so defined has useful mathematical properties.	Accomplished in the current research.
Can a similar definition be provided for other sampling and aggregation schemes?	Probably.	Requires more research.
If so, are the textures of the same picture function under two different schemes comparable?	Unknown.	Requires more research.
Can a similar definition of texture be provided for continuous picture functions independent of sampling?	Unknown.	Requires more research.
Is texture related to the notion of fractal functions?	Probably.	This relates to the previous question.

Question	Answer	Comment
Does texture as defined on GBT grids correspond to our human idea of visual texture?	Preliminary, yes.	Accomplished in the current research.
How can textural similarities and dissimilarities be measured?	By clustering in space and probably other methods.	Spatial clustering was done in this research. More work is required.
Can a computer be programmed to automatically decompose an image into regions of similar texture?	Probably.	Requires more research.
If so, will such a decomposition be similar to a decomposition provided by a person?	Probably.	Requires much research and experimentation.
If it is achieved, would such a decomposition be suited for further processing to derive shapes, features, and relations of entities in the image?	Yes.	The decomposition process would be directed at that goal. However, much research and experimentation is required.

Table 11. Questions and Answers about Research on Texture.

The main need is to proceed forward from the current base to achieve techniques for automated image decomposition into regions of similar texture. Other issues relating to quantifying texture in a more general setting also need to be examined.

6.2 Shape

During this research we developed and showed the properties of a powerful shape descriptor called f^2. We further showed how f^2 and other shape descriptors could be used in conjunction with the GBT cellular hierarchy to analyze shape over regions of varying size. The current situation is that we have developed powerful tools but have not yet learned how to use them well. The most important area of future research is to learn how to use the current tools efficiently to solve a variety of image understanding problems. Other significant areas are the deriving of new shape descriptors and the development of a knowledge data base against which descriptions of entities in images can be compared.

Table 12 shows the status of various questions regarding needed follow on work on shape descriptors.

Question	Answer	Comment
Are there functions which are useful in describing the shape of a picture function within a region?	Yes. Many are classical and one was developed in this research.	The f^2 function and its properties are described in this report.
Could there be other useful functions for describing shape?	Almost certainly.	Only second moments were used to develop f^2. Use of higher moments should yield other functions.
Do we know how to use the current functions to best advantage?	No. We have only preliminary results.	Much research is required.
Do the current functions have any utility independently from the GBT system?	Yes. In the current research only the texture vector is GBT dependent.	GBT aggregates are a natural vehicle for computing picture function descriptors.
Are shape descriptors all that is needed to understand entities within images?	No. A knowledge data base is necessary for understanding.	The current work provides only for the understanding of simple geometric shapes. Much further research is required.
Can the shape descriptors in GBT cells be extended to arbitrary regions?	Almost certainly. A successful first effort was made during this research.	More research in this area is needed.

Table 12. Questions and Answers about Research on Shape Descriptors.

7.0 BIBLIOGRAPHY

1. R. Duda and P. Hart, Pattern Classification and Scene Analysis, New York: John Wiley & Sons, Inc., 1973

2. L. Gibson and C. Lenzmeier, A Hierarchical Pattern Extraction System for Hexagonally Sampled Images, Final Report on Contract Number F49620-81-C-0039, October 1981.

3. L. Gibson and D. Lucas, "Spatial Data Processing Using Generalized Balanced Ternary," Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing, June 1982.

4. L. Gibson and D. Lucas, "Vectorization of Raster Images using Hierarchical Methods," Computer Graphics and Image Processing, Volume 20, No. 1, September 1982.

5. D. Lucas, "A Multiplication in N-Space," Proceedings of the American Mathematical Society, Volume 74, Number 1, April 1979.

6. D. Lucas and L. Gibson, A System for Hierarchical Addressing in Euclidean Space, Martin Marietta Corporation, January 1980.

7. D. Lucas and L. Gibson, "Automated Data Capture," Proceedings of the Symposium on Automation Technology for Management and Productivity Advancement through CAD/CAM and Data Handling, November 1982.

8. A. Rosenfeld and A. C. Kak, Digital Picture Processing, New York: Academic Press, 1976.

8.0 APPENDIX. GENERALIZED BALANCED TERNARY

WHAT IS GBT?

GBT is a method of representing a two dimensional surface that enables a computer to work easily with data distributed on that surface. When a person looks at data in two dimensions, a map sheet or a photographic image, for example, he quickly sees the general content of that map or photograph, and can examine certain aspects of the content in detail if he chooses to focus his attention. Computers which use conventional representations for planar data have difficulty performing this simple chore. Such systems permit easy examination of the smallest components of the data, be they pixels, bits, bytes, or vectors, but have no efficient mechanisms for examining the data in aggregate form. As the saying goes, they cannot see the forest for the trees.

The GBT method of structuring a surface permits data to be aggregated so that an automated system can examine the general content of the data without looking at the detail. Because of this capability, an algorithm can determine at a high level if the generalized information suffices for the algorithm's purpose or whether finer information is needed. In the latter case, the GBT structure permits the layered accessing of finer and finer data until the finest granularity is reached. This capability for selective access to successive levels of detail is taken for granted in human perception, but it has presented a significant problem in designing systems for machine perception. It is fundamental, for example, to the notions of scene analysis, feature extraction, and pattern recognition. The GBT system solves this problem.

GBT is not only a method for representing space, it is also an addressing scheme that allows access to that representation. Further, it contains an algebraic system which operates on the addressing scheme. These aspects of GBT will be described below. A GBT structure can be implemented in any dimension, however, a discusson of dimensions other than the second is beyond the scope of this paper. GBT stands for Generalized Balanced Ternary, a term signifying that GBT is the higher dimensional analogue of the one dimensional system known as balanced ternary.

THE STRUCTURE AND ADDRESSES

The GBT structure is one of a hierarchy of cells. At each level, the cells are constructed of cells from the previous level according to a rule of aggregation. The basic cells of this structure are hexagons. Figure 1 shows the hexagonal covering of the plane. This covering has the uniform adjacency property, that is, each element of the covering is adjacent to exactly six other elements and shares with each exactly one-sixth of its boundary. In contrast, a covering of the plane with squares does not have uniform adjacencies. Some squares are adjacent at a point while others share a side.

A first level aggregate is formed by taking a single hexagon and its six neighbors (see Figure 2a). The first level aggregates also cover the plane and have the uniform adjacency property. In general, an aggregate at level n is formed by taking a level n-1 aggregate and its six neighbors. It can be shown that the planar covering and uniform adjacency properties hold at each level. Figures 2b and 2c show second and third level aggregates.

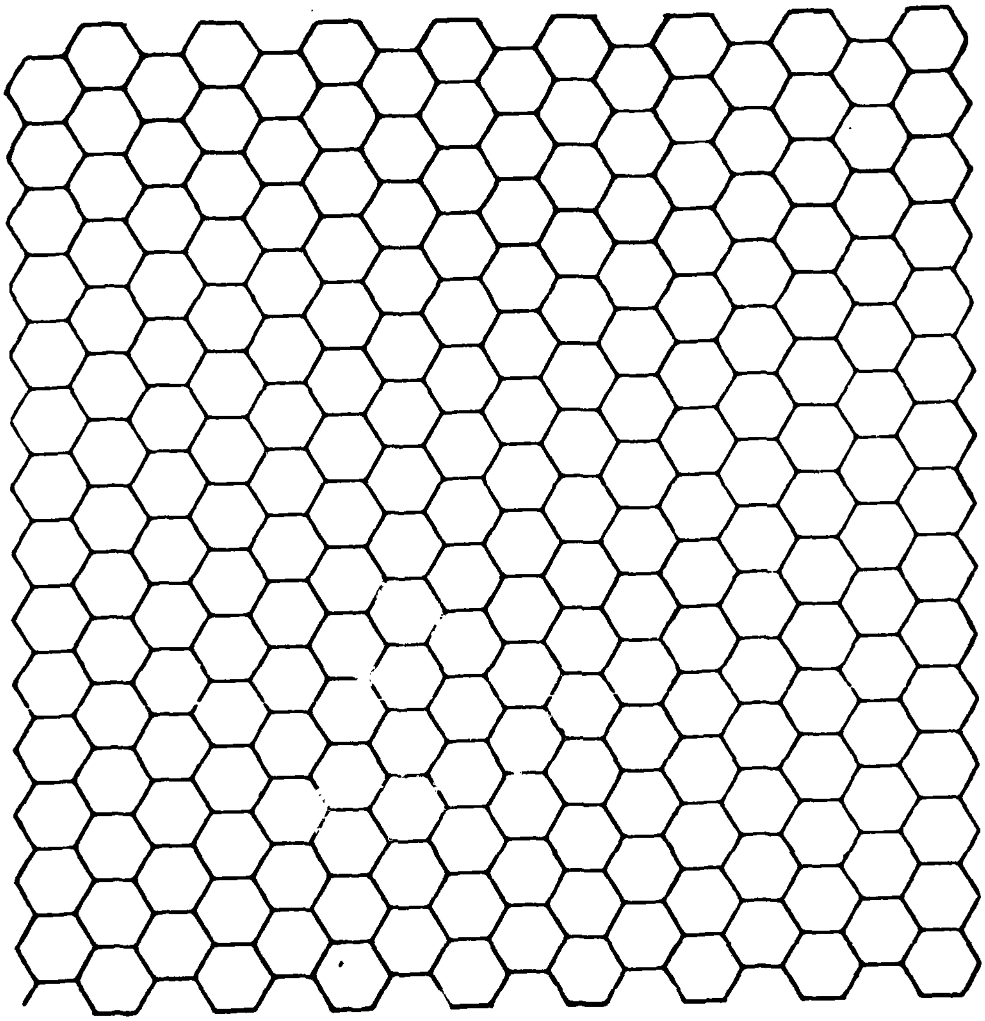
The GBT addressing system is based on the following scheme. In an aggregate, the center cell is labeled 0 and the outer six cells are labeled, in clockwise order, 1, 3, 2, 6, 4, 5 (see Figure 2). Each hexagon in the plane has a unique GBT address, a sequence of digits corresponding to the labels of the cells above that hexagon.

Each digit of the address corresponds to an aggregate level. For example, the address 536 labels the hexagon in the 6 position of the first level aggregate, which is in the 3 position of the second level aggregate, which is in the 5 position of the third level aggregate, which is at the 0 or center position at all higher levels. The hexagon 536 is shaded in Figure 3.

The digit 7 is used to address entire aggregates rather than hexagons. Thus, the first level aggregate shaded in Figure 4 has address 117. The second level aggregate outlined in the same figure has address 677.

The symbols 0, 1, 2, 3, 4, 5, 6, and 7, used as GBT digits, are also used in octal and decimal notation to express integers. Using them in this new context allows GBT addresses to be handled directly by computer hardware and software. Care must be taken, however, not to confuse the addresses with integers.

FIGURE 1: The Hexagonal Covering of the Plane



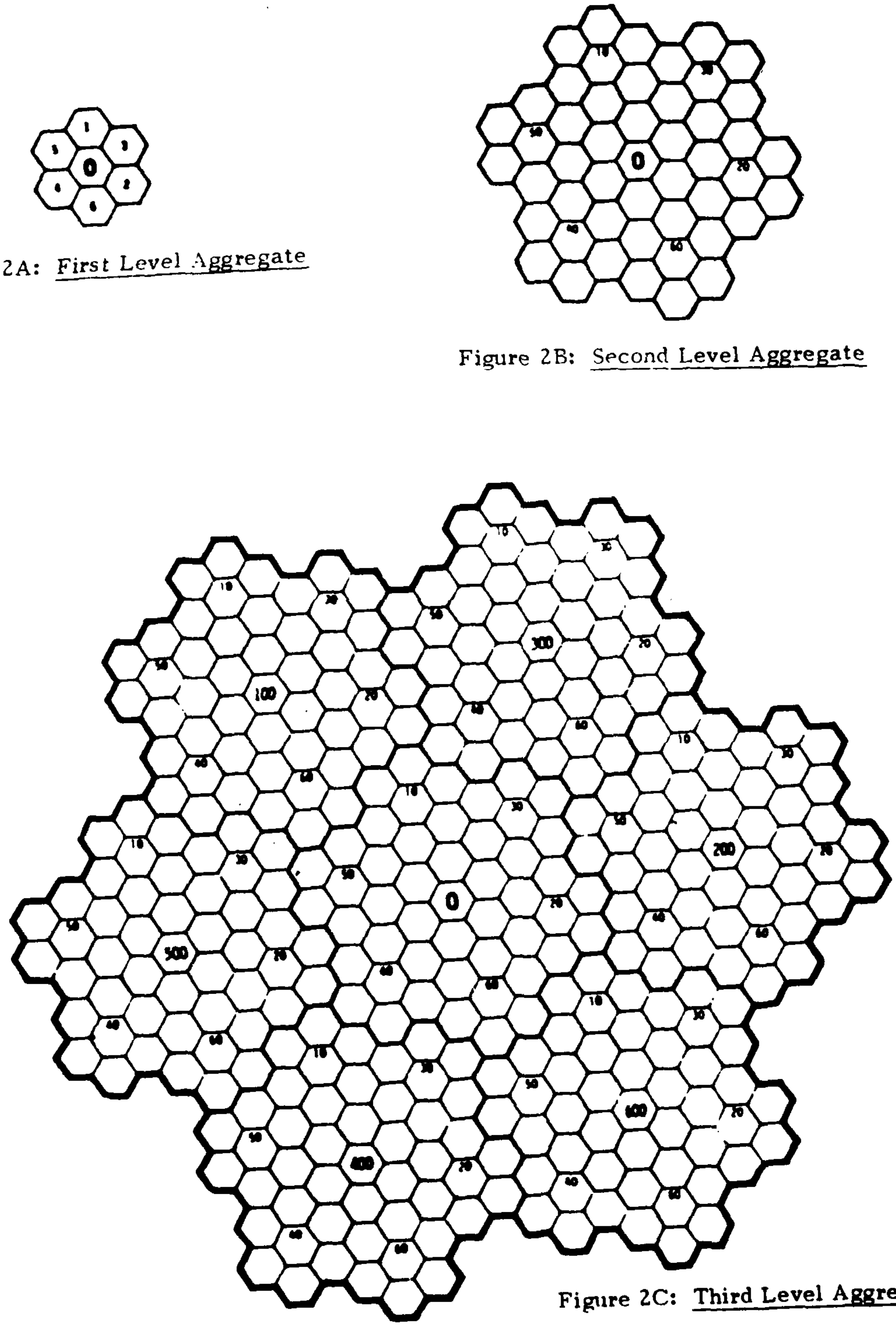


Figure 2A: First Level Aggregate

Figure 2B: Second Level Aggregate

Figure 2C: Third Level Aggregate

FIGURE 2: The Aggregate Structure

FIGURE 3: The Location of the Hexagon

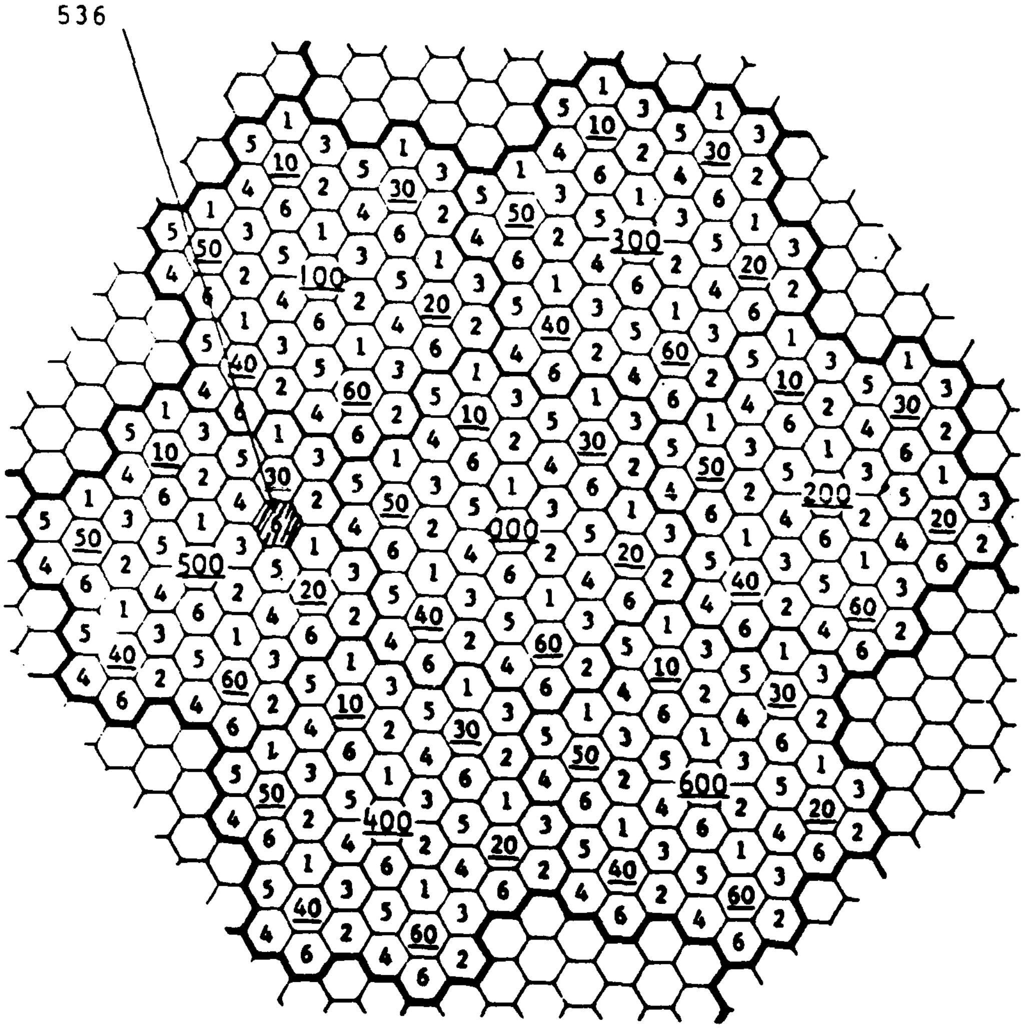
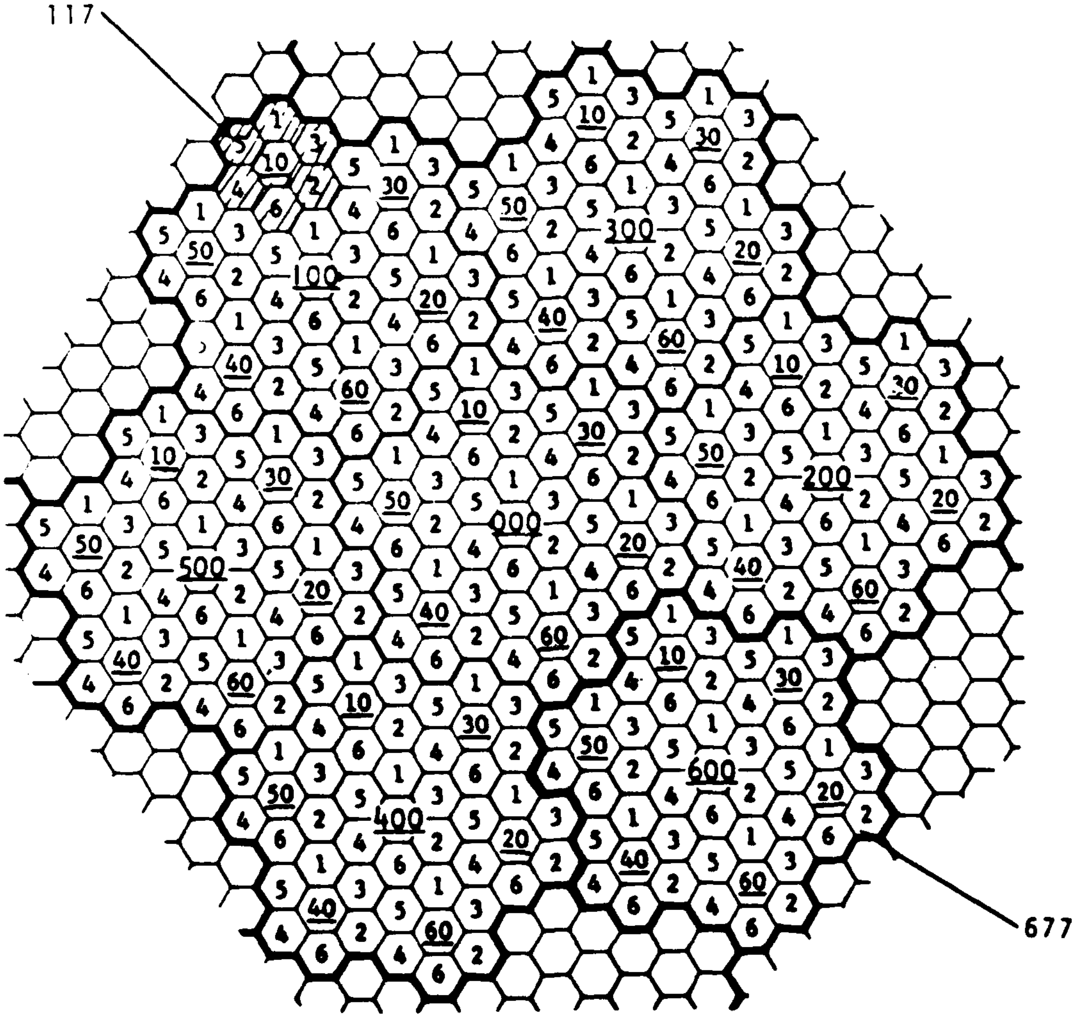


FIGURE 4:

Two Examples of the Use of the Failing Sevens Notation
to Address Higher Aggregates



THE ARITHMETIC

In order for the GBT addressing system to be useful, there must be efficient methods for doing planar addition, subtraction, and multiplication entirely in terms of GBT addresses. These methods are discussed below:

1. Addition: Addition of GBT addresses parallels integer addition in that if an addition table for the seven digits is given, any two addresses can be added. From Figure 5, we can derive the basic GBT addition table. Using standard planar parallellogram addition, we see, for example, that 1+2=3, 3+6=2, and 5+6=4. If 3 and 2 are added, the sum is outside the central first level aggregate, 3+2=25. Thus, Table 1 is obtained. Addition of multidigit addresses is very much like adding multidigit integers. For example (see Figure 6), to add 153 and 45, add 3+5=1, then 5+4=52, and carrying the 5 to the next column, 5+1=16. Thus, 153+45=1621. Figure 6B shows these vectors in the plane.

2. Subtraction: Subtraction is accomplished by the process of complementing and adding. The complement of a GBT address is its digitwise sevens complement. The complement of 61542 is 16235, for example. In Figure 5, we see that 3 and 4 are complements, as are 1 and 6, and 53 and 24.

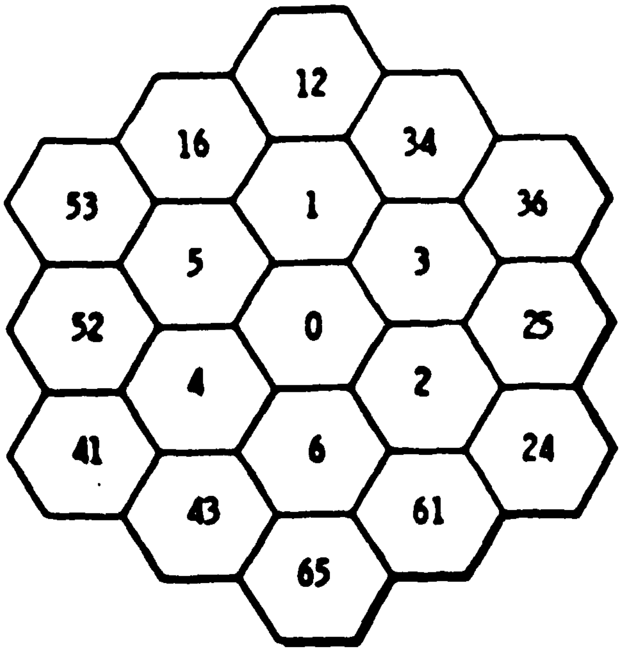
3. Multiplication: GBT multiplication is similar to addition in that it is a digitwise operation. The multiplication table (Table 2) shows that the GBT product of two digits is just their integer product modulo 7. An example of multidigit multiplication is:

   254 x 62:

   ```
     254
   x  62
   -----
     431    (= 2 x 254)
    523     (= 6 x 254)
   -----
    5261    (= the GBT sum)
   ```

   Figure 7 illustrates this product geometrically.

FIGURE 5: Key to Planar Addition Table



+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	12	3	34	5	16	0
2	2	3	24	25	6	0	61
3	3	34	25	36	0	1	2
4	4	5	6	0	41	52	43
5	5	16	0	1	52	53	4
6	6	0	61	2	43	4	65

TABLE 1: GBT Addition

Figure 6A

Add the Addresses 153 and 45:

```
(1)(5)            ( ) Denotes a Carry
   153
    45
  ____
  1621
```

Figure 6B

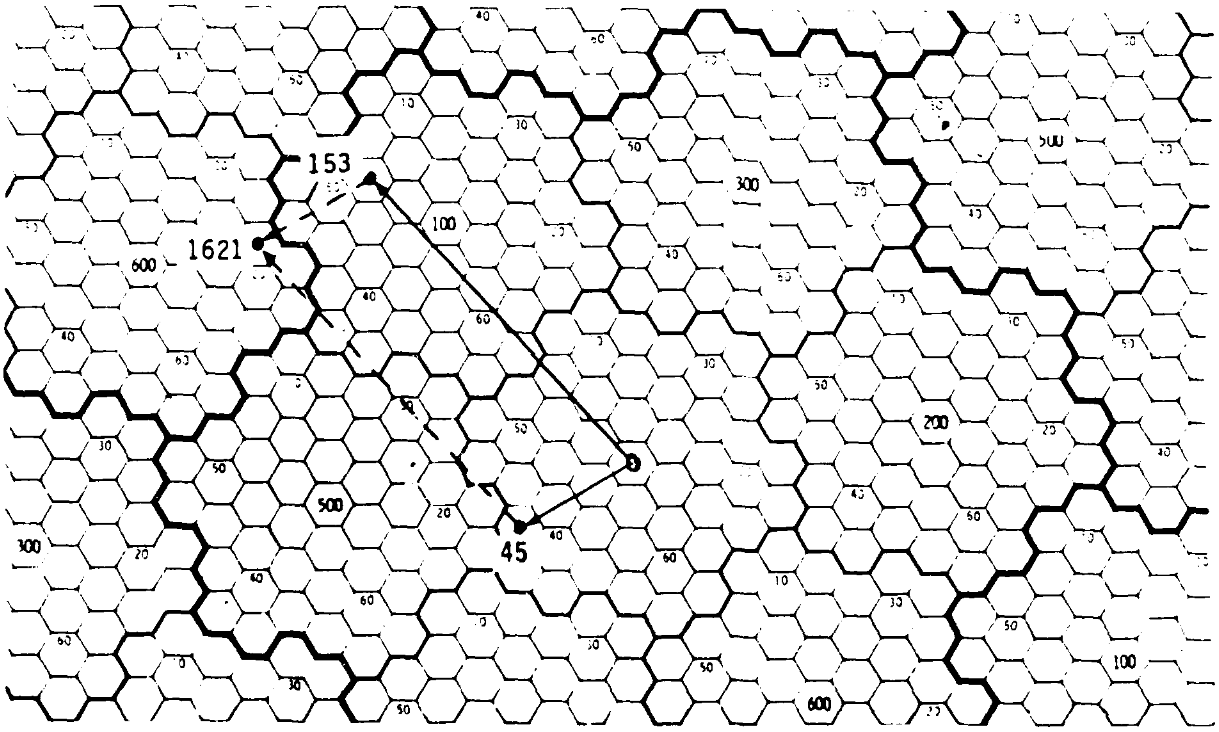
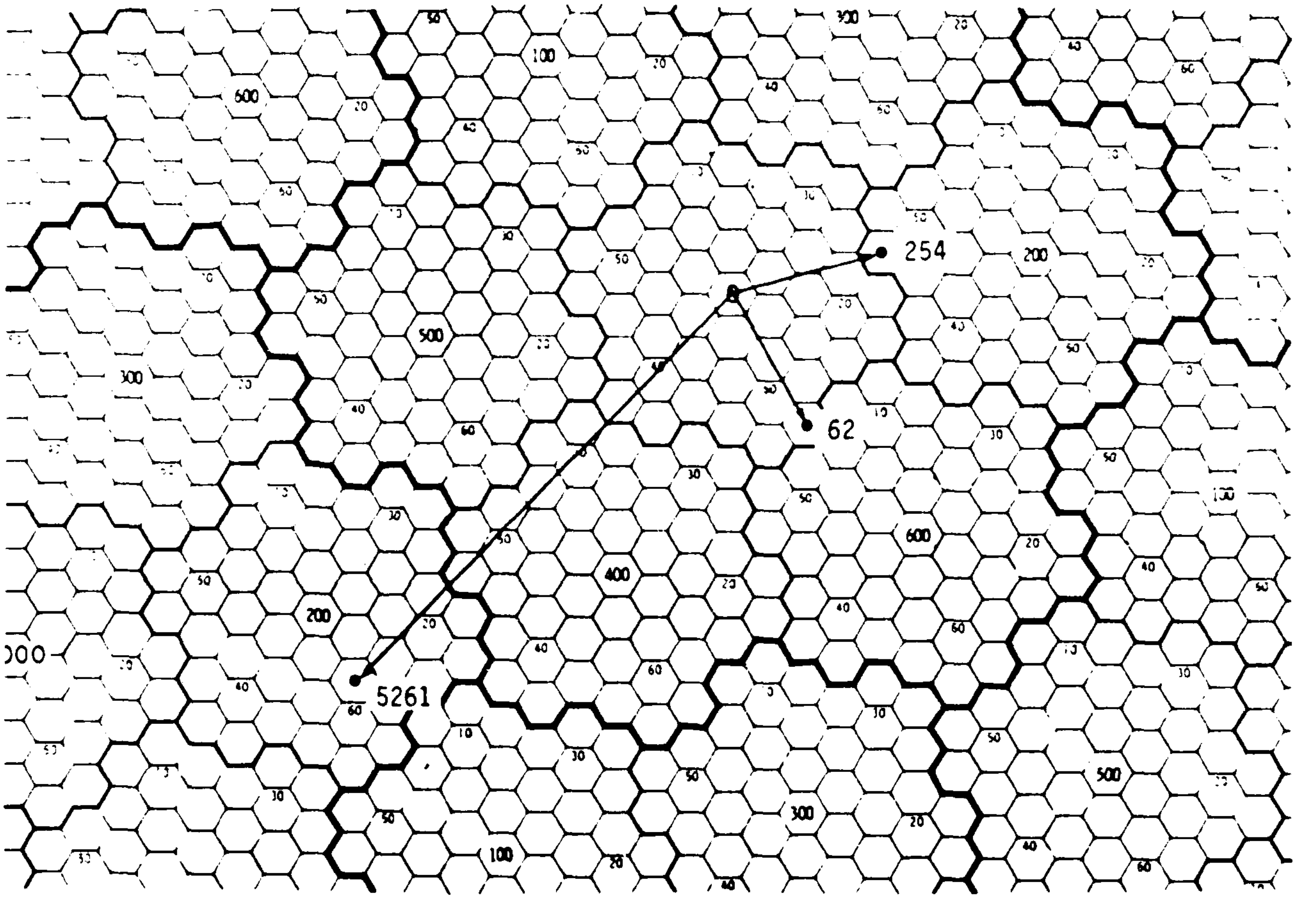


FIGURE 6: The GBT Sum of 153 and 45

•	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

TABLE 2: GBT Multiplication

FIGURE 7: The GBT Product: 254 x 62



THE GBT DATA BASE STRUCTURE

Any set of GBT addresses generates a tree structure. The tree derived from a list of 8 four digit GBT addresses is shown in Figure 8. This tree originates from a root node which corresponds to the universal cell containing all addressable cells. Attached to the root node is a node corresponding to the first of a set consisting of those next lower level cells beneath the universal cell which contain any of the cells on the list. Generally, each node in the tree corresponds to a cell in the GBT cellular hierarchy. Every node within a particular level of the tree structure is linked to its siblings in sorted order by the digit of the GBT address corresponding to the level. The first node within the list is attached to the parent cell for that level. Any given node can have, at most, seven subordinates, but may have fewer.

This tree structure provides an easy way of accessing data by general location. Figure 9 shows the location of the cells whose addresses are listed in Figure 8. Suppose someone were to ask for a list of the addresses stored in the tree which are within four units (1 unit is the distance between hexagon centers) of 4221. A simple computation shows that any such address must be a subordinate of one of the cells 4277, 6577, 6477, and 43177. The latter cell is not part of the four digit universe used in this example, and therefore, is not checked. To check the contents of 4277, we start at the tree root, follow a pointer to cell 4777, then to cell 4277. Then we follow all pointers from the 4277 node to discover the addresses 4201, 4207, 4245, and 4267. These are placed on a list of candidate addresses. To check 6477 and 6577, we first descend from the root node to cell 4777 then to the node corresponding to 6777. At the cell 6777 information is stored within the cell describing the subordinates to the cell. It is at this point we discover that there is no pointer to 6477, so there can be no stored addresses subordinate to that cell. We then follow the pointer from cell 6777 to cell 6077 and from there to cell 6017 and finally to cell 6577. By following all subsequent pointers from cell 6577 we locate cells 6562 and 6566 which are added to the list of candidate addresses. The next step is to use the GBT subtraction operation and vector length measurement to eliminate those addresses on the list which are more than four units from 4221. This step eliminates 4245 and reduces the list to 4201, 4207, 4267, 6562, and 6566, all of which are (at least partially) within 4 units of 4221.

FIGURE 8:

A List of Addresses and the Corresponding Tree Structure

Nodes in the tree corresponding to addresses on the list are circled.

ADDRESSES: 4201, 4207, 4245, 4267, 6023, 6177, 6562, 6566

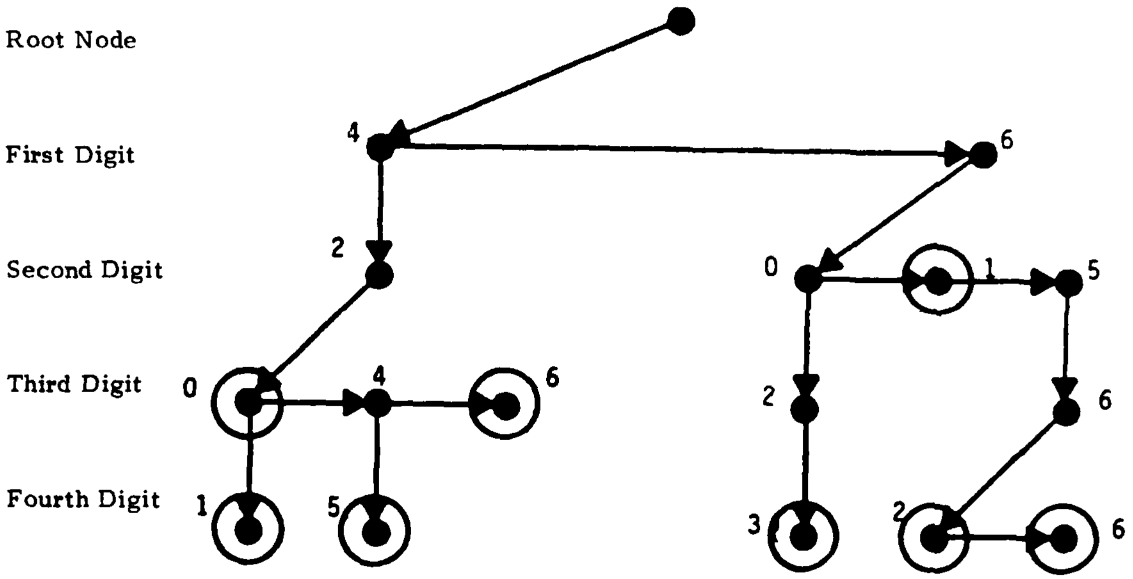
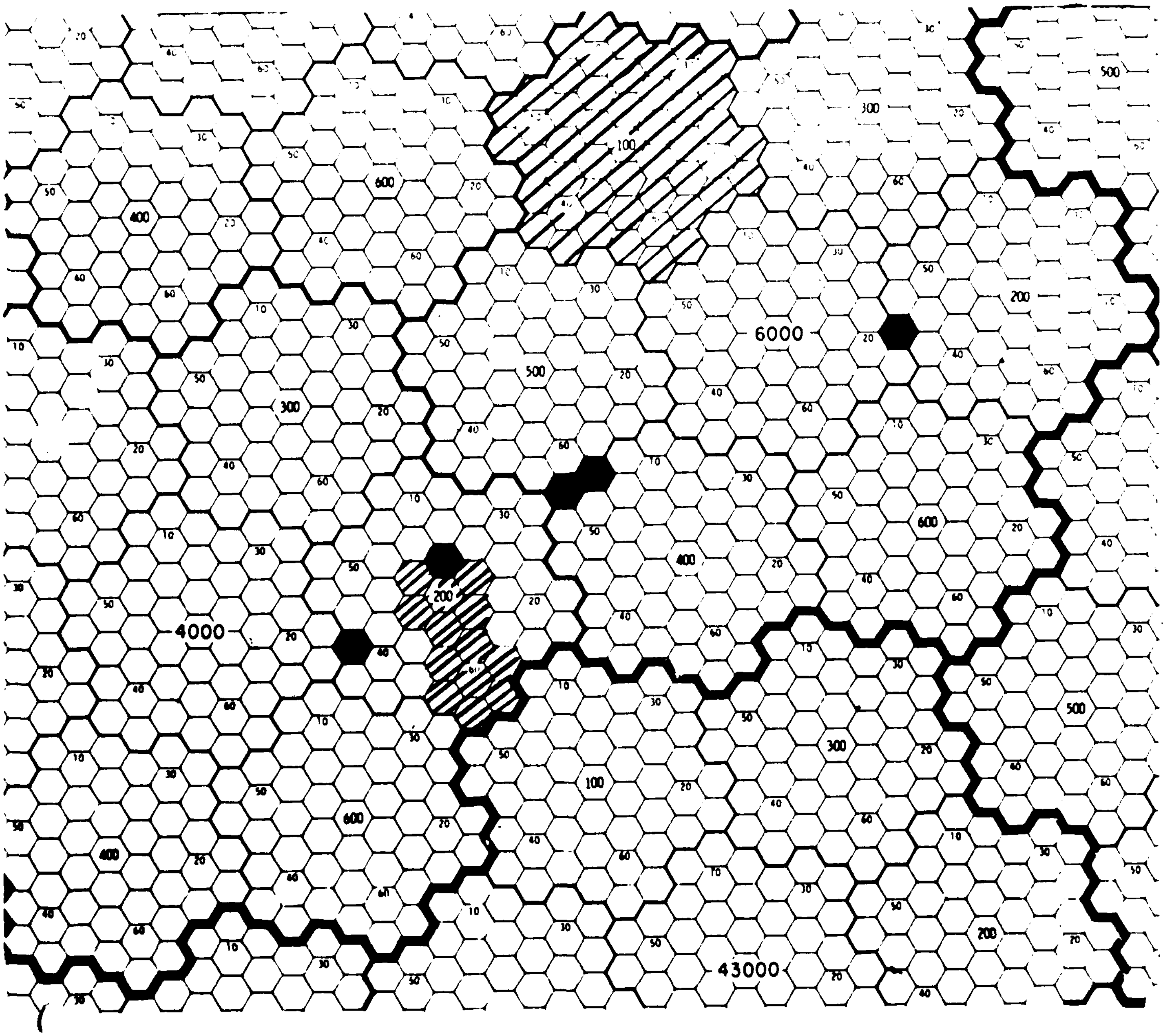


FIGURE 9: The Locations of Cells on the Address List



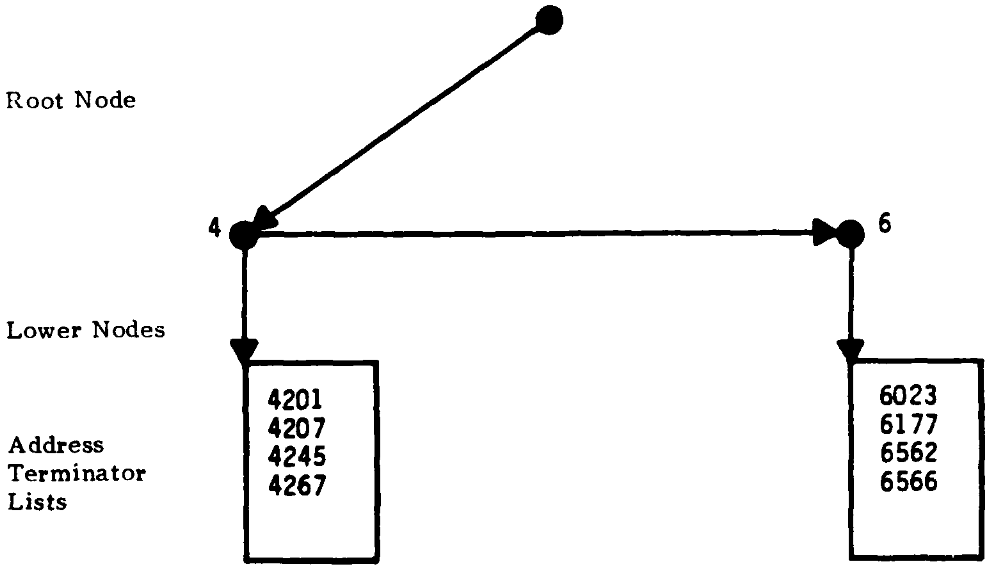
General GBT data accesses operate in the same manner as the above examnle. If a region R is to be searched, the addresses of a small number of large cells which cover R are determined. Then these cells are searched by examining their subordinate nodes in the data structure until all stored subordinates of these cells are accessed. Then, this list is checked against the boundary of region R to determine which cells actually intersect that region. This is a relatively direct procedure for accessing by location in that it avoids any consideration of addresses which lie outside the set of cells which cover R. This locational access technique is not of great utility if the number of addresses stored in the data structure is small. However, in large data bases, maps or imagery, for instance, which contain multiple millions of points, some such access mechanism is imperative to rapid operation.

The tree structure shown in Figure 8, and described above, is wasteful in that some nodes exist in the tree solely to provide a linkage between a larger cell and a very small number (1 or 2) of subordinates. These problems can be overcome by the structure indicated in Figure 10. This abbreviated tree structure does not contain separate nodes for many of the parent cells of addresses which are stored. Instead, the suffix digits of a stored address, or stored addresses, are listed in a node corresponding to an address which may be several levels above the stored addresses in the hierarchy. A new level in the abbreviated tree structure is created only when the number of cells below the parent cells becomes large enough that it is no longer efficient to use list processing methods for the cells. The effect of the abbreviated tree structure is that stored addresses may be accessed by passing through fewer nodes than in the full tree structure of Figure 8. The locational access properties of the full tree are preserved in the abbreviated version.

FIGURE 10:

An Abbreviated Tree Corresponding to the same Address List as Figure 8

The address terminator lists shown in boxes are pointed to by the node just above them.



END

DATE
FILMED

4-83

DTIC